On the Expected Size of Conformal Prediction Sets

Guneet S. Dhillon
Department of Statistics
University of Oxford
guneet.dhillon@stats.ox.ac.uk

George Deligiannidis
Department of Statistics
University of Oxford
deligian@stats.ox.ac.uk

Tom Rainforth
Department of Statistics
University of Oxford
rainforth@stats.ox.ac.uk

Abstract

While conformal predictors reap the benefits of rigorous statistical guarantees on their error frequency, the size of their corresponding prediction sets is critical to their practical utility. Unfortunately, there is currently a lack of finite-sample analysis and guarantees for their prediction set sizes. To address this shortfall, we theoretically quantify the expected size of the prediction sets under the split conformal prediction framework. As this precise formulation cannot usually be calculated directly, we further derive point estimates and high-probability interval bounds that can be empirically computed, providing a practical method for characterizing the expected set size. We corroborate the efficacy of our results with experiments on real-world datasets for both regression and classification problems.

1 INTRODUCTION

Imagine a company that recognizes a market where multiple businesses are interested in the same task, e.g., the inspection and quality control of manufactured goods. Seeing this opportunity, the company decides to provide an AI-powered solution, to help automate and streamline the process for these customers.

A major characteristic that customers would want to know is: *how often will the system make errors?* Moreover, they will often want to constrain the frequency of errors to a level acceptable for their particular use case. The conformal prediction framework [Vovk et al., 2005; Shafer and Vovk, 2008] fulfills this requirement. Instead of a single label, it predicts a set of labels (based on past experiences) and rigorously guarantees that the error of the predicted set not containing the true label is bound to a user-specified level. Notably, the split conformal prediction framework [Papadopoulos et al., 2002; Vovk et al., 2005; Lei et al., 2015] further provides computational efficiency for practical deployment.

A natural follow-up question that customers might ask is: *how big are the prediction sets expected to be?* This is a valid concern. For instance, a naive prediction of the entire label space achieves zero error; however, such predictions are not useful. Thus, the sizes of the prediction sets are of significant relevance in practice.

Existing works have considered the asymptotic behavior of the expected size of prediction sets, analyzing conformal predictors in the context of statistical optimality [Lei et al., 2013; Lei and Wasserman, 2013; Vovk et al., 2014; Lei, 2014; Lei et al., 2015; Vovk et al., 2016; Sadinle et al., 2019]. In practice, however, we are concerned with prediction set sizes in the finite-sample setting. In fact, in most practical applications of conformal prediction—such as image classification [Angelopoulos et al., 2021; Fisch et al., 2021a], natural language processing [Fisch et al., 2021b,a; Schuster et al., 2021], drug discovery [Fisch et al., 2021b,a], clinical trials [Lu and Kalpathy-Cramer, 2021; Lu et al., 2022], robotics [Dixit et al., 2023; Lindemann et al., 2023], and election polling [Cherian and Bronner, 2020]—algorithms are compared based on: (i) their frequencies of error, and (ii) the expected sizes of their prediction sets.

Currently, this expected size is empirically estimated by averaging the sizes of the constructed prediction sets over multiple runs, via Monte Carlo averaging. For our hypothetical company and customers, this amounts to the company collecting labeled data from multiple customers and running predictions repeatedly; this is an expensive procedure. Additionally, each customer will have a different set of parameters—e.g., different frequency of error requirements, different number of labeled data, etc.—each resulting in a different expected set size. To provide a satisfactory answer to each customer, the company will have to run this procedure

Proceedings of the 27$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

separately for each set of values. As such, it is often not feasible to provide customers with an indication of the expected prediction set size using this approach.

To overcome these shortcomings, we theoretically quantify the expected size of split conformal prediction sets. As computing this quantity directly is often intractable in practice, we derive procedures to empirically approximate it. Our proposed procedures require data to be collected *only once* to provide a point estimate and high-probability interval bounds for the expected prediction set size. Consequently, our hypothetical company no longer requires access to labeled data from multiple customers and repeated runs of the conformal algorithm. Instead, the company could use a single set of pre-collected in-house data to compute both point and interval estimates for the expected size of the prediction sets constructed by its proposed (conformal) system. From the customer's perspective, this information allows them to reliably evaluate the company's system and further determine whether to use it or not.

In summary, our contributions are as follows:

- We theoretically quantify the expected size of the prediction sets constructed under the split conformal prediction framework (cf. Section 4).
- We derive practical point estimates and high-probability intervals for the above (cf. Section 5).
- We illustrate the efficacy of our results experimentally on regression and classification (cf. Section 6).

2 BACKGROUND

We are concerned with supervised learning, where we are provided with labeled data and want to predict the label for new test inputs. The split conformal prediction framework [Papadopoulos et al., 2002; Vovk et al., 2005; Lei et al., 2015] predicts a set of labels for a given test input such that the probability of error, i.e., the predicted set not containing the true label, is guaranteed to be bound at a user-specified level. This is achieved by first splitting the labeled data into training and calibration data. The training data trains a (non-conformity) scoring function, which is used to compute scores on the calibration data; then a threshold is determined using these calibration scores. The algorithm uses the computed score threshold to construct the prediction set for a new test input.

Formally, we denote $Z_i = (X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$, for $i = 1, \ldots, n+1$, to be $n+1$ data points sampled i.i.d. from an arbitrary distribution over feature and label spaces $\mathcal{X}$ and $\mathcal{Y}$ respectively. We treat $Z_{1:n} = \{Z_1, \ldots, Z_n\}$ as the calibration data and Z_{n+1} as the test datum; the training data is used to implement the scoring function as discussed later. For a significance level $\alpha \in (0, 1)$, we want to predict a set of labels $\hat{C}_\alpha(X_{n+1}; Z_{1:n}) \subseteq \mathcal{Y}$ such that the probability of error is bound by α, i.e.,

$$\mathbb{P}\left\{Y_{n+1} \notin \hat{C}_\alpha\left(X_{n+1}; Z_{1:n}\right)\right\} \leq \alpha. \tag{1}$$

This is a marginal probability, taken over both the test datum Z_{n+1} and the calibration data $Z_{1:n}$.

The split conformal framework achieves this by reasoning about a non-conformity function [Vovk et al., 2005; Shafer and Vovk, 2008]. We denote this function by $R : \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{R}$, which maps a data point to a real-valued non-conformity score ($\mathcal{R} \subseteq \mathbb{R}$). This function quantifies how non-conforming a data point is. If $R(x, y)$ is small, then (x, y) is conforming; conversely, if $R(x, y)$ is large, then (x, y) is non-conforming or atypical. This function is implemented as a distance function L between the label y and its prediction $\hat{y} = M(x)$ obtained from a machine learning model M. The model M is trained on training data that is independent of the test and the calibration data. Then, the non-conformity score is defined as $R(x, y) = L(M(x), y)$. For example, for regression problems, the model M could be a deep neural network, and the loss L the l_1 loss, resulting in the non-conformity score $R(x, y) = |M(x) - y|$.

The split conformal framework uses the above function to compute non-conformity scores. For each calibration datum Z_i, for $i = 1, \ldots, n$, we denote its corresponding calibration non-conformity score as $R_i = R(Z_i)$. Since the test datum label Y_{n+1} is not observed, we consider all possible realizations $y \in \mathcal{Y}$ of it and denote the corresponding test non-conformity score as $R(X_{n+1}, y)$, a function of the random variable X_{n+1} and the fixed variable y. Subsequently, the framework computes an acceptance score threshold using only the calibration data; this is denoted by $\tau_\alpha(R_{1:n})$ and is set to the $\lceil(1-\alpha)(n+1)\rceil$'th smallest value in the augmented set of calibration scores $\{R_1, \ldots, R_n, \infty\}$. Then, the label y is included in the prediction set if its corresponding test non-conformity score is below this acceptance threshold, i.e., the split conformal prediction set is defined as,

$$\hat{C}^R_\alpha(X_{n+1}; Z_{1:n}) = \{y \in \mathcal{Y} : \tau_\alpha(R_{1:n}) \geq R(X_{n+1}, y)\}. \tag{2}$$

These prediction sets satisfy Equation (1) [Papadopoulos et al., 2002; Vovk et al., 2005; Lei et al., 2015].

Note that there is a naive way of satisfying the constraint in Equation (1). Consider the predictor that predicts the entire space of labels; this satisfies the constraint, but the predictions are uninformative. Therefore, *the size of the prediction sets plays an important role in a conformal predictor's efficacy*—the smaller the size of the prediction sets, the better the predictor.

3 RELATED WORK

Since its original development [Gammerman et al., 1998; Saunders et al., 1999], many works have built on the basic conformal prediction framework. Notably, Papadopoulos et al. [2002]; Vovk et al. [2005]; Lei et al. [2015] proposed the aforementioned split conformal prediction as a special case. We refer readers to Angelopoulos and Bates [2023] for a comprehensive overview.

The work done on conformal prediction set sizes has mostly focused on the statistical optimality of the family of predictors defined under this framework. Such works use the expected size of the prediction sets as a notion of *inefficiency* for the predictors—the smaller the better. They show that conformal predictors are *optimal* under different settings—such as unsupervised learning [Lei et al., 2013, 2015], regression [Lei and Wasserman, 2013], binary classification [Lei, 2014], and multi-class classification [Sadinle et al., 2019]—by showing that the expected size of their prediction sets asymptotically converges to that of an oracle. Additionally, Vovk et al. [2014, 2016]; Sadinle et al. [2019] provide similar optimality results when the probability of error is constrained conditionally per class/label.

This notion of viewing the expected size of the prediction sets as an inefficiency has propagated to practical settings as well, where conformal predictors are compared based on their average empirical prediction set sizes. Furthermore, different non-conformity functions have been proposed to reduce this quantity. For instance, Romano et al. [2019]; Kivaranovic et al. [2020] propose using quantile regression to train the machine learning model and an associated quantile interval loss function, Sadinle et al. [2019]; Romano et al. [2020]; Angelopoulos et al. [2021] propose loss functions for classification based on the predicted class/label probabilities, and Bellotti [2021]; Stutz et al. [2022] learn the non-conformity function in an end-to-end fashion by making the conformal pipeline differentiable.

4 THEORETICAL QUANTIFICATION

We are interested in analyzing the size of prediction sets under the split conformal framework. Similar to previous works, we will analyze the expected prediction set size $\mathbb{E}[|\hat{C}_\alpha^R(X_{n+1}; Z_{1:n})|]$.[1] While theoretical works considered the asymptotic behavior of this quantity, empirical works want to estimate it in practice. We aim to bridge the two by theoretically quantifying the expected set size in the finite-sample case and deriving procedures to estimate it in practice, providing useful practical information about the prediction sets.

We begin with the quantification. The prediction set size is the reference measure of the set of labels included. However, from Equation (2) we know that the prediction set depends on: (i) the test non-conformity scores $R(X_{n+1}, y)$, for all labels $y \in \mathcal{Y}$, (ii) the calibration non-conformity scores $R_{1:n}$, and (iii) the significance level α. This implies that the prediction set is dependent on the test datum feature and the calibration data only through their corresponding non-conformity scores. Therefore, instead of considering the label space, we analyze the space of non-conformity scores.

As a result, we compute the reference measure of the set of non-conformity scores below the acceptance score threshold $\tau_\alpha(R_{1:n})$, as determined by the framework. To translate this measure back to the label space, we introduce a multiplicative factor $\#_R(r)$, which we discuss in detail later. With the multiplicative factor $\#_R(r)$ and the acceptance score threshold $\tau_\alpha(R_{1:n})$, we show in Theorem 1 that the expected prediction set size is,

$$\mathbb{E}\left[\left|\hat{C}_\alpha^R(X_{n+1}; Z_{1:n})\right|\right] = \int_{\mathcal{R}} \mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq r\right\} \#_R(r) dr, \tag{3}$$

where the probability $\mathbb{P}\{\tau_\alpha(R_{1:n}) \geq r\}$ is over the calibration data $Z_{1:n}$ and is not easy to compute.

To simplify it further, we assume that the calibration non-conformity scores are i.i.d. from a probability distribution with the corresponding probability density/mass function p_R.[2] Note that we are given i.i.d. calibration data $Z_{1:n}$, so their corresponding non-conformity scores $R_{1:n}$ are i.i.d. as well; here we define the distribution they follow. With this, the individual probabilities of each calibration score being strictly smaller than r are identical. We denote this by $\tilde{P}_R(r)$ and define it as,

$$\tilde{P}_R(r) = \mathbb{P}\left\{R_1 < r\right\} = \int_{\mathcal{R}} \mathbb{1}\left\{r' < r\right\} p_R(r')\, dr', \tag{4}$$

noting that this is similar to, but not the same as, the cumulative distribution function $P_R(r) = \mathbb{P}\{R_1 \leq r\}$.

As we are concerned with the event that the acceptance score threshold, i.e., the $\lceil (1-\alpha)(n+1) \rceil$'th smallest calibration score, is larger than r, we can allow at most $n_\alpha = \lceil (1-\alpha)(n+1) \rceil - 1$ calibration scores to be strictly less than r. In doing so, we can now express $\mathbb{P}\{\tau_\alpha(R_{1:n}) \geq r\}$ as the cumulative distribution function of a binomial random variable. The binomial random variable has n trials and success probability $\tilde{P}_R(r)$, which we denote by $B(n, \tilde{P}_R(r))$. The required cumulative distribution function is evaluated at n_α, and

[1] We overload $|\cdot|$ and $\int_{\mathcal{Y}} dy$ to be the Lebesgue measure for continuous and the counting measure for discrete spaces.

[2] For a random variable X, we use p_X and P_X to denote the probability density/mass function and the cumulative distribution function respectively.

Table 1: **Multiplicative factor under different settings.** We summarize the multiplicative factor under different settings that we will utilize in Section 6 for the experiments. Most settings utilize $M(x)$, the model prediction for input x; for LAC [Sadinle et al., 2019], $M_y(x)$ is the predicted probability for label y and for CQR [Romano et al., 2019], $M_\beta(x)$ is the prediction for the β'th level quantile and $M_\Delta(x) = (M_{1-\alpha/2}(x) - M_{\alpha/2}(x))/2$. Details of these settings and the derivations of their multiplicative factors are provided in Appendix B.

Problem type	Loss function	Non-conformity function $R(x, y)$	Multiplicative factor $\#_R(r)$
Regression ($\mathcal{Y} = \mathbb{R}$)	l_1	$\lvert M(x) - y\rvert$	2
	$l_{p\geq 1}$	$\lvert M(x) - y\rvert^p$	$2r^{1/p-1}/p$
	CQR [Romano et al., 2019]	$\max\{M_{\alpha/2}(x) - y,\ y - M_{1-\alpha/2}(x)\}$	$\begin{cases} 2, & r \geq 0 \\ 2\left(1 - P_{M_\Delta(X_{n+1})}(-r)\right), & r < 0 \end{cases}$
Classification (discrete $\mathcal{Y}$)	0-1	$\mathbb{1}\{M(x) \neq y\}$	$\begin{cases} 1, & r = 0 \\ \lvert\mathcal{Y}\rvert - 1, & r = 1 \end{cases}$
	LAC [Sadinle et al., 2019]	$1 - M_y(x)$	$\sum_{y\in\mathcal{Y}} p_{M_y(X_{n+1})}(1 - r)$

is denoted by $P_{B(n,\tilde{P}_R(r))}(n_\alpha)$. Therefore, the expected prediction set size in Equation (3) simplifies to,

$$\mathbb{E}\left[\left|\hat{C}^R_\alpha(X_{n+1}; Z_{1:n})\right|\right] = \int_{\mathcal{R}} P_{B(n,\tilde{P}_R(r))}(n_\alpha)\, \#_R(r) dr. \tag{5}$$

We package these two results together in the following theorem and provide the proof in Appendix A.1.

Theorem 1 (Expected size of prediction sets)**.** *If the test and the calibration non-conformity scores are independent of each other, then the expected size of the split conformal prediction sets is given by Equation* (3)*. Furthermore, if the calibration non-conformity scores are i.i.d., then the expected size is given by Equation* (5)*.*

These results pertain to the marginal expected set size. We include conditional expectations in Section 4.2.

Multiplicative Factor The multiplicative factor is responsible for translating the reference measure on the space of non-conformity scores to the reference measure on the label space. Formally, we define this factor as,

$$\#_R(r) = \int_{\mathcal{Y}} p_{R(X_{n+1},y)}(r)\, dy, \tag{6}$$

where $p_{R(X_{n+1},y)}$ is the probability density/mass function of the random variable $R(X_{n+1}, y)$, with the randomness from the random variable X_{n+1} (the test datum feature) and not the fixed variable y (a label).

We provide derivations of the multiplicative factor under different settings in Appendix B and summarize them in Table 1. A common loss for regression is the l_1 loss, with the multiplicative factor 2. We generalize to any l_p loss ($p \geq 1$), with the multiplicative factor $2r^{1/p-1}/p$; this highlights the reference measure translation that the multiplicative factor performs. The 0-1 loss is a candidate for classification, with the multiplicative factor 1 and $|\mathcal{Y}|-1$ for $r = 0$ and $r = 1$ respectively. Alternatively, more nuanced non-conformity functions have multiplicative factors that depend on the distribution of data and the machine learning model used.

For instance, Sadinle et al. [2019] propose the least ambiguous set-valued classifiers (LAC) with the non-conformity function $R(x, y) = 1 - M_y(x)$, where $M_y(x)$ is the predicted probability for label y. The associated multiplicative factor (cf. Table 1) cannot be analytically solved without making assumptions about the data distribution and/or the machine learning model. However, LAC provably constructs prediction sets with minimum expected size if the predicted probabilities are correct; this does not hold in practice, but the predicted sets are small. Similarly, conformalized quantile regression (CQR, Romano et al. [2019]) for regression and adaptive prediction sets (APS, Romano et al. [2020]) for classification construct small prediction sets, but their associated multiplicative factors are intractable without further assumptions. This does not come as a surprise; the split conformal framework satisfies Equation (1), but the quality of the prediction sets constructed depends on the data distribution and the machine learning model used [Vovk et al., 2005; Shafer and Vovk, 2008].

We do not wish to make additional assumptions, so we treat the multiplicative factor associated with such non-conformity functions as unknown. Note that Theorem 1 holds for any choice of the non-conformity function.

4.1 Insights

After quantifying the expected prediction set size in Theorem 1, we analyze its dependence on various user-specified parameters, providing general insights into influencing the quantity from a user's perspective. We empirically validate our analysis in Appendix C.4.

Non-conformity Function The non-conformity function (constituting the machine learning model and the loss function) plays an important role. Its influence on the set size is through the binomial random variable's success probability $\tilde{P}_R(r)$ and the multiplicative factor $\#_R(r)$ (cf. Equation (5)). For example, for regression, l_1 and CQR [Romano et al., 2019] use different machine learning models (with CQR using quantile regression models) and different loss functions (l_1 versus CQR loss). For classification, 0-1, LAC [Sadinle et al., 2019], and APS [Romano et al., 2020] are different loss functions that could be used atop the same machine learning model. Such modifications alter $\tilde{P}_R(r)$ and $\#_R(r)$ and hence the expected set size (cf. Section 6).

There are scenarios where the non-conformity function can change, but the multiplicative factor does not. E.g., under the l_1 loss for regression, one can change the machine learning model, but $\#_R(r) = 2$ remains the same (cf. Table 1). In such cases, only the influence through the binomial random variable's success probability $\tilde{P}_R(r)$ matters. Consider $\tilde{P}_{R_1}$ and $\tilde{P}_{R_2}$ corresponding to two such non-conformity functions R_1 and R_2 respectively, where the non-conformity score distribution of the first first-order stochastically dominates the second, i.e., $\tilde{P}_{R_1}(r) \leq \tilde{P}_{R_2}(r)$, for all $r \in \mathcal{R}$. Consequently, the expected set size is larger for the first function, $\mathbb{E}[|\hat{C}_\alpha^{R_1}(X_{n+1}; Z_{1:n})|] \geq \mathbb{E}[|\hat{C}_\alpha^{R_2}(X_{n+1}; Z_{1:n})|]$. Therefore, for small expected set sizes, with the multiplicative factor being the same, $\tilde{P}_R$ should be skewed to have most of its probability density/mass on small values of $r \in \mathcal{R}$. A common recipe to achieve this is by using a machine learning model that generalizes well. As an analytical tool, a practitioner could also plot the empirical distribution of the calibration non-conformity scores to compare different non-conformity functions.

When the multiplicative factor changes, it is not straightforward to compare different non-conformity functions without making further assumptions. This is especially the case when the space of non-conformity scores is modified in the process. E.g., for regression, the l_1 loss admits non-negative non-conformity scores only, whereas CQR [Romano et al., 2019] additionally permits negative scores. If we were able to enforce and/or assume that CQR's scores smaller than $-c$ (for some constant $c \geq 0$) are never permissible, we could translate its space of non-conformity scores to the set of non-negative reals $\mathbb{R}_{\geq 0}$ by adding an offset of c. Then, offsetted CQR's multiplicative factor would never be larger than that of the l_1 loss (cf. Table 1); additionally, if the l_1 loss non-conformity score distribution first-order stochastically dominates that of offsetted CQR's scores, CQR would achieve a smaller expected set size. Comparisons under other modifications of the multiplicative factor would require different assumptions.

Significance Level The significance level is generally fixed by a user rather than being tunable, specifying the user's requirement on the frequency of error. However, we use this as a sanity check and highlight the trade-off between the error and the size of the prediction sets. Intuitively, as the significance level increases, the framework allows for more errors which decreases the size of its prediction sets. Indeed, in Equation (5), an increase in α causes a decrease in n_α, which further prompts a decrease in the expected prediction set size.

Number of Calibration Data Labeled data procurement is often difficult, and a user might need justification for the benefits of collecting more data. When using it for calibration, it is unclear how it would influence the expected prediction set size. In Equation (5), an increase in n causes an increase in both the number of trials of the binomial random variable and the value at which the cumulative distribution function of the said random variable is evaluated; the former decreases the expected set size while the latter increases it, diminishing their contributions. Resolving this disagreement would require making more assumptions about the distribution of the calibration non-conformity scores.

4.2 Conditional Expectation

We quantified the marginal expected prediction set size in Theorem 1, where we marginalize over the randomness induced by both the test datum feature and the calibration data. For instance, when advertising its prediction system to potential customers, our company in Section 1 computes the marginal expected set size as there is no additional information to condition on. However, customers might be interested in evaluating the quality of the prediction sets constructed on a particular test input. In this case, the marginal expected set size is not the quantity of interest, but the conditional expected set size conditioned on that test datum is. We therefore extend Theorem 1 to allow for conditional expectations of the prediction set size.

When conditioning on the test datum feature $X_{n+1} = x_{n+1}$, there is additional information about the distribution of $R(X_{n+1}, y)$ and hence the multiplicative factor. Synonymous with our definition of the multiplicative factor in Equation (6), we introduce the feature-specific multiplicative factor $\#_R(r; x_{n+1}) = \int_{\mathcal{Y}} \delta_{R(x_{n+1},y)}(r) dy$, where the probability density/mass function $p_{R(X_{n+1},y)}$ is replaced by $\delta_{R(x_{n+1},y)}$, the Dirac delta distribution that places all of its probability mass on $R(x_{n+1}, y)$. Then, the conditional expected set size is given by,

$$\begin{aligned} &\mathbb{E}\left[\left|\hat{C}_\alpha^R(X_{n+1}; Z_{1:n})\right| \middle| X_{n+1} = x_{n+1}\right] \\ &= \int_{\mathcal{R}} \mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq r\right\} \#_R\left(r; x_{n+1}\right) dr, \end{aligned} \tag{7}$$

and if the calibration non-conformity scores are i.i.d.,

$$= \int_{\mathcal{R}} P_{B\left(n,\tilde{P}_R(r)\right)}\left(n_\alpha\right) \#_R\left(r; x_{n+1}\right) dr. \tag{8}$$

Note that Equations (7) and (8) are analogous to the marginals in Equations (3) and (5) respectively. We summarize this below (with the proof in Appendix A.2).

Corollary 2 (Expected size of prediction sets conditioned on the test datum feature)**.** *If the test and the calibration non-conformity scores are independent of each other, then the expected size of the split conformal prediction sets conditioned on the test datum feature $X_{n+1} = x_{n+1}$ is given by Equation* (7)*. Furthermore, if the calibration non-conformity scores are i.i.d., then the conditional expected size is given by Equation* (8)*.*

We can similarly condition on the calibration data $Z_{1:n} = z_{1:n}$; we summarize the result in Corollary 4.

5 PRACTICAL ESTIMATION

We theoretically quantify the expected prediction set size in Theorem 1. However, it assumes knowledge of the multiplicative factor $\#_R(r)$ and the binomial random variable's success probability $\tilde{P}_R(r)$, for all non-conformity scores $r \in \mathcal{R}$. While the former may be known under some settings (cf. Table 1), the latter is unknown in most practical scenarios as it relies on the distribution of the calibration non-conformity scores.

Currently, the expected set size is empirically estimated by averaging the size of the constructed prediction sets over multiple runs, i.e., a Monte Carlo average. This equates to sampling a (pseudo) calibration data, obtaining conformal prediction sets on multiple (pseudo) test data, and repeating the process many times. The average size of the obtained sets will estimate the expected set size; if repeated enough times, this estimate would be close to the true value. For our company in Section 1, this involves collecting large amounts of labeled data from its customers and repeatedly executing the above procedure. Furthermore, each such estimation scheme is instantiated with a fixed configuration of the significance level and the number of calibration data, resulting in an estimate that is configuration-specific. Therefore, the company will need to carry out this Monte Carlo averaging scheme numerous times to obtain satisfactory estimates for varying values of the parameters. This becomes infeasible in practice.

Alternatively, knowing the quantification of the expected prediction set size from Theorem 1, we can develop procedures to estimate the value directly. This will require data to be collected *only once*; we will assume access to $Z'_1 = (X'_1, Y'_1), \ldots, Z'_k = (X'_k, Y'_k)$, k data points drawn i.i.d. from the data distribution.

Going back to our hypothetical company, possible ways of obtaining this data are either from a customer or held-out in-house company data. We will detail procedures to derive point and interval estimates for the expected prediction set size using these accessible data points. Our goals in doing so are: (i) for the point estimate to be close to the expected set size, and (ii) for the interval to bound the expected set size with high probability. We provide a summary in Table 2.

Note that we consider the marginal expected set size, but our procedures can extend to the conditionals by substituting in the conditionally given quantities.

5.1 Known Multiplicative Factor

We begin with the setting where the multiplicative factor can be analytically calculated and is known. We compute the non-conformity scores for the k accessible data points as $R'_i = R(Z'_i)$, for $i = 1, \ldots, k$. We further use these non-conformity scores to empirically approximate $\tilde{P}_R(r)$, for all $r \in \mathcal{R}$, with the quantity,

$$\tilde{P}_R^{\text{emp}}(r) = \frac{1}{k} \sum\nolimits_{i=1}^{k} \mathbb{1}\left\{R'_i < r\right\}. \tag{9}$$

By replacing $\tilde{P}_R(r)$ with $\tilde{P}_R^{\text{emp}}(r)$ in Equation (5), we obtain a point estimate for the expected set size. $\tilde{P}_R^{\text{emp}}$ can also be used to estimate the expected set size under different significance levels and number of calibration data as $\tilde{P}_R$ is not dependent on these parameters.

We further provide guarantees for this estimate. We use the work of Dvoretzky et al. [1956]; Massart [1990] to bound the difference between a cumulative distribution function and its empirical approximation. Specifically, we can compute a $1-\gamma$ confidence interval for $\tilde{P}_R(r)$ of the form $[\tilde{P}_R^{\text{emp}}(r) - \Delta_{k,\gamma}, \tilde{P}_R^{\text{emp}}(r) + \Delta_{k,\gamma}]$, where $\Delta_{k,\gamma} = \sqrt{\ln(2/\gamma)/2k}$. Thus by replacing $\tilde{P}_R(r)$ with $\tilde{P}_R^{\text{emp}}(r) \pm \Delta_{k,\gamma}$ in Equation (5), we obtain the lower-upper bounds corresponding to a $1-\gamma$ confidence interval for the expected set size. This is summarized in the following result (with the proof in Appendix A.3).

Corollary 3 (Confidence interval for the expected prediction set size)**.** *Following Equation* (5) *(Theorem 1), with a known multiplicative factor, the expected size of split conformal prediction sets lies in the interval,*

$$\left[\int_{\mathcal{R}} P_{B\left(n,\tilde{P}_R^{emp}(r)+\Delta_{k,\gamma}\right)}\left(n_\alpha\right) \#_R(r) dr, \int_{\mathcal{R}} P_{B\left(n,\tilde{P}_R^{emp}(r)-\Delta_{k,\gamma}\right)}\left(n_\alpha\right) \#_R(r) dr\right], \tag{10}$$

with probability at least $1-\gamma$*.*

As k increases, the error term $\Delta_{k,\gamma}$ decreases and so does the width of the confidence interval. Therefore,

Table 2: **Practical estimates under different settings.** We summarize our point and interval estimates derived when the multiplicative factor is known (cf. Section 5.1) and when it is unknown (cf. Section 5.2).

Setting	**Our point estimate**	**Our interval estimate** (with significance γ)
Known multiplicative factor	$\int_{\mathcal{R}} P_{B\left(n, \tilde{P}_R^{\text{emp}}(r)\right)}\left(n_\alpha\right) \#_R(r) dr$	$\Big[\int_{\mathcal{R}} P_{B\left(n, \tilde{P}_R^{\text{emp}}(r)+\Delta_{k,\gamma}\right)}\left(n_\alpha\right) \#_R(r) dr,$ $\int_{\mathcal{R}} P_{B\left(n, \tilde{P}_R^{\text{emp}}(r)-\Delta_{k,\gamma}\right)}\left(n_\alpha\right) \#_R(r) dr\Big]$
Unknown multiplicative factor	$\int_{\mathcal{Y}} \frac{1}{k} \sum_{i=1}^{k} P_{B\left(n, \tilde{P}_R^{\text{emp}}\left(R(X_i', y)\right)\right)}\left(n_\alpha\right) dy$	$\Big[\int_{\mathcal{Y}} \frac{1}{k} \sum_{i=1}^{k} P_{B\left(n, \tilde{P}_R^{\text{emp}}\left(R(X_i', y)\right)+\Delta_{k,\gamma}\right)}\left(n_\alpha\right) dy,$ $\int_{\mathcal{Y}} \frac{1}{k} \sum_{i=1}^{k} P_{B\left(n, \tilde{P}_R^{\text{emp}}\left(R(X_i', y)\right)-\Delta_{k,\gamma}\right)}\left(n_\alpha\right) dy\Big]$

the larger k is, the tighter the confidence interval gets. In fact, as $k \to \infty$, the error term $\Delta_{k,\gamma} \to 0$ and the confidence interval contracts to our point estimate.

5.2 Unknown Multiplicative Factor

Next, we consider the setting where the multiplicative factor is intractable due to its dependence on the data distribution and/or the machine learning model. One way to get around this is to estimate the factor using density estimation methods and substitute in its value.

To provide a self-contained approach, we re-arrange the formulation of the expected set size in Equation (5) to get rid of the multiplicative factor (cf. Appendix A.5). Instead, the quantification contains the expectation over the random variable $R(X_{n+1}, y)$ as follows,

$$\begin{aligned} &\mathbb{E}\left[\left|\hat{C}_\alpha^R(X_{n+1}; Z_{1:n})\right|\right] \\ &= \int_{\mathcal{Y}} \mathbb{E}\left[P_{B\left(n, \tilde{P}_R(R(X_{n+1}, y))\right)}\left(n_\alpha\right)\right] dy. \end{aligned} \tag{11}$$

The expectation $\mathbb{E}[P_{B(n, \tilde{P}_R(R(X_{n+1}, y)))}(n_\alpha)]$ contains two unknowns: $\tilde{P}_R$ inside the expectation, and the distribution of $R(X_{n+1}, y)$ over which the expectation is evaluated. This can be empirically approximated using nested Monte Carlo methods [Rainforth et al., 2018] with the accessible data points: we approximate $\tilde{P}_R$ with $\tilde{P}_R^{\text{emp}}$ (cf. Equation (9)), and the distribution of $R(X_{n+1}, y)$ with the samples $R(X_i', y)$, for $i = 1, \ldots, k$. This amounts to the approximation $\frac{1}{k} \sum_{i=1}^{k} P_{B(n, \tilde{P}_R^{\text{emp}}(R(X_i', y)))}(n_\alpha)$ for the expectation term. Integrating this quantity over $y \in \mathcal{Y}$ results in a point estimate for the expected set size, as desired.

Additionally, we can compute intervals synonymous with Equation (10) by replacing $\tilde{P}_R^{\text{emp}}(R(X_i', y))$ with $\tilde{P}_R^{\text{emp}}(R(X_i', y)) \pm \Delta_{k,\gamma}$ above. However, these may not be valid confidence intervals due to the extra approximation; we refer readers to Rainforth et al. [2018] for nested Monte Carlo estimates' error analysis. Despite this, we demonstrate their practical utility in Section 6.

6 EXPERIMENTS

We now illustrate the efficacy of our results experimentally by applying our estimation procedures derived in Section 5 on real-world datasets from the UCI database [Kelly et al., 2023]. We use the l_1 loss and CQR [Romano et al., 2019] non-conformity functions for regression, and the 0-1 loss, LAC [Sadinle et al., 2019], and APS [Romano et al., 2020] for classification. We include the main experimental results here, with additional ones incorporated in Appendices C and D. Our code is available at https://github.com/Guneet-Dhillon/expected-conformal-prediction-set-size.

6.1 Experimental Setup

Similar to Tibshirani et al. [2019], we randomly split a dataset into 25% training, 25% calibration, and 50% test. We use the training data to train a random forest [Breiman, 2001] for the non-conformity function, utilizing the scikit-learn [Pedregosa et al., 2011] implementation with 100 trees.[3] We run the split conformal algorithm on the calibration and the test data, with the significance level α set to 0.1. We repeat this process 1000 times, where we sample new training data every 100 runs, and new calibration and test data every run.

6.2 Marginal Expected Prediction Set Size

We begin with the marginal expected size of prediction sets; we compare our derived estimates with the commonly used Monte Carlo averaging in approximating the marginal expected set size. The latter is the average size of the prediction sets constructed using the split conformal algorithm on the test data using the calibration data, across all data splits. For the former, we need accessible data points to make approximations. While any data drawn i.i.d. from the data distribution suffices, we use the calibration data as the accessible data to facilitate direct comparison of the two esti-

[3]For CQR [Romano et al., 2019], we train a quantile regression forest [Meinshausen, 2006] using the implementation from https://github.com/zillow/quantile-forest.

Table 3: **Marginal expected prediction set sizes.** We illustrate the marginal expected sizes of split conformal prediction sets using different non-conformity functions and UCI datasets. The estimates are obtained via Monte Carlo averaging, our point estimates, and our interval estimates (lower-upper bounds with $\gamma = 0.1$). We also compute the absolute error between our individual point estimates and the mean Monte Carlo average. We report the means and standard deviations. For classification, the number of classes/labels is provided in parentheses.

			Marginal expected prediction set size				
		Dataset	Our interval lower bound	Monte Carlo average	Our point estimate	Our interval upper bound	Absolute error
Regression ($\mathcal{Y} = \mathbb{R}$)	l_1	Abalone	$1.87_{0.07}$	$2.19_{0.09}$	$2.19_{0.09}$	$2.71_{0.12}$	$0.07_{0.05}$
		AirFoil	$1.11_{0.07}$	$1.39_{0.10}$	$1.39_{0.09}$	$2.03_{0.13}$	$0.08_{0.05}$
		AirQuality	$0.01_{0.00}$	$0.02_{0.00}$	$0.02_{0.00}$	$0.02_{0.00}$	$0.00_{0.00}$
		BlogFeedback	$2.30_{0.02}$	$2.38_{0.02}$	$2.38_{0.02}$	$2.47_{0.02}$	$0.02_{0.01}$
		CTSlices	$0.17_{0.01}$	$0.18_{0.01}$	$0.18_{0.01}$	$0.20_{0.01}$	$0.01_{0.00}$
		FacebookComments	$0.38_{0.01}$	$0.41_{0.01}$	$0.41_{0.01}$	$0.44_{0.01}$	$0.01_{0.00}$
		OnlineNews	$2.77_{0.03}$	$2.91_{0.03}$	$2.91_{0.03}$	$3.07_{0.03}$	$0.03_{0.02}$
		PowerPlant	$0.64_{0.01}$	$0.70_{0.02}$	$0.70_{0.02}$	$0.78_{0.02}$	$0.01_{0.01}$
		Superconductivity	$0.92_{0.02}$	$1.02_{0.02}$	$1.02_{0.02}$	$1.13_{0.03}$	$0.02_{0.01}$
		WhiteWineQuality	$2.29_{0.06}$	$2.58_{0.08}$	$2.58_{0.07}$	$2.99_{0.09}$	$0.06_{0.05}$
	CQR [Romano et al., 2019]	Abalone	$1.81_{0.06}$	$2.17_{0.98}$	$2.16_{0.17}$	$2.44_{0.05}$	$0.15_{0.09}$
		AirFoil	$1.36_{0.05}$	$1.58_{0.64}$	$1.58_{0.07}$	$2.09_{0.13}$	$0.05_{0.04}$
		AirQuality	$0.02_{0.00}$	$0.02_{0.11}$	$0.02_{0.00}$	$0.02_{0.00}$	$0.00_{0.00}$
		BlogFeedback	$1.39_{0.01}$	$1.39_{0.96}$	$1.39_{0.01}$	$1.39_{0.01}$	$0.01_{0.01}$
		CTSlices	$0.28_{0.01}$	$0.28_{0.50}$	$0.28_{0.01}$	$0.28_{0.01}$	$0.01_{0.00}$
		FacebookComments	$0.55_{0.04}$	$0.56_{2.27}$	$0.55_{0.04}$	$0.55_{0.04}$	$0.03_{0.03}$
		OnlineNews	$2.88_{0.02}$	$2.96_{0.77}$	$2.96_{0.02}$	$3.07_{0.03}$	$0.02_{0.01}$
		PowerPlant	$0.69_{0.01}$	$0.73_{0.26}$	$0.73_{0.01}$	$0.79_{0.01}$	$0.01_{0.01}$
		Superconductivity	$0.79_{0.01}$	$0.82_{0.71}$	$0.82_{0.01}$	$0.85_{0.01}$	$0.01_{0.01}$
		WhiteWineQuality	$2.24_{0.10}$	$2.24_{0.89}$	$2.24_{0.10}$	$2.27_{0.10}$	$0.07_{0.06}$
Classification (discrete $\mathcal{Y}$)	0-1	APSFailure (2)	$1.00_{0.00}$	$1.00_{0.00}$	$1.00_{0.00}$	$1.00_{0.00}$	$0.00_{0.00}$
		Adult (2)	$2.00_{0.00}$	$2.00_{0.00}$	$2.00_{0.00}$	$2.00_{0.00}$	$0.00_{0.00}$
		Avila (12)	$1.00_{0.00}$	$1.00_{0.00}$	$1.00_{0.00}$	$1.00_{0.00}$	$0.00_{0.00}$
		BankMarketing (2)	$1.00_{0.00}$	$1.00_{0.00}$	$1.01_{0.02}$	$1.74_{0.23}$	$0.01_{0.02}$
		CardDefault (2)	$2.00_{0.00}$	$2.00_{0.00}$	$2.00_{0.00}$	$2.00_{0.00}$	$0.00_{0.00}$
		Landsat (6)	$1.05_{0.21}$	$4.79_{2.14}$	$4.47_{1.31}$	$6.00_{0.01}$	$1.07_{0.82}$
		LetterRecognition (26)	$1.00_{0.00}$	$1.00_{0.00}$	$1.00_{0.01}$	$6.50_{5.85}$	$0.00_{0.01}$
		MagicGamma (2)	$1.97_{0.06}$	$2.00_{0.00}$	$2.00_{0.00}$	$2.00_{0.00}$	$0.00_{0.00}$
		SensorLessDrive (11)	$1.00_{0.00}$	$1.00_{0.00}$	$1.00_{0.00}$	$1.00_{0.00}$	$0.00_{0.00}$
		Shuttle (7)	$1.00_{0.00}$	$1.00_{0.00}$	$1.00_{0.00}$	$1.00_{0.00}$	$0.00_{0.00}$
	LAC [Sadinle et al., 2019]	APSFailure (2)	$0.91_{0.01}$	$0.93_{0.26}$	$0.93_{0.01}$	$0.93_{0.00}$	$0.00_{0.01}$
		Adult (2)	$1.09_{0.01}$	$1.11_{0.32}$	$1.11_{0.01}$	$1.14_{0.01}$	$0.01_{0.00}$
		Avila (12)	$0.91_{0.00}$	$0.93_{0.26}$	$0.93_{0.01}$	$0.95_{0.01}$	$0.00_{0.00}$
		BankMarketing (2)	$0.96_{0.00}$	$0.99_{0.12}$	$0.99_{0.00}$	$1.01_{0.01}$	$0.00_{0.00}$
		CardDefault (2)	$1.20_{0.01}$	$1.25_{0.44}$	$1.25_{0.01}$	$1.32_{0.01}$	$0.01_{0.01}$
		Landsat (6)	$0.96_{0.01}$	$1.02_{0.25}$	$1.02_{0.02}$	$1.10_{0.02}$	$0.01_{0.01}$
		LetterRecognition (26)	$0.94_{0.01}$	$0.97_{0.32}$	$0.97_{0.01}$	$1.02_{0.01}$	$0.01_{0.00}$
		MagicGamma (2)	$1.03_{0.01}$	$1.07_{0.26}$	$1.07_{0.01}$	$1.12_{0.01}$	$0.01_{0.01}$
		SensorLessDrive (11)	$0.90_{0.00}$	$0.91_{0.29}$	$0.91_{0.00}$	$0.92_{0.00}$	$0.00_{0.00}$
		Shuttle (7)	$0.99_{0.00}$	$0.99_{0.12}$	$0.99_{0.00}$	$0.99_{0.00}$	$0.00_{0.00}$
	APS [Romano et al., 2020]	APSFailure (2)	$0.91_{0.00}$	$0.92_{0.33}$	$0.92_{0.00}$	$0.93_{0.00}$	$0.00_{0.00}$
		Adult (2)	$1.20_{0.01}$	$1.23_{0.50}$	$1.23_{0.01}$	$1.26_{0.01}$	$0.01_{0.00}$
		Avila (12)	$1.15_{0.02}$	$1.22_{0.69}$	$1.22_{0.02}$	$1.29_{0.03}$	$0.02_{0.01}$
		BankMarketing (2)	$1.07_{0.01}$	$1.09_{0.46}$	$1.09_{0.01}$	$1.11_{0.01}$	$0.01_{0.00}$
		CardDefault (2)	$1.30_{0.01}$	$1.36_{0.50}$	$1.36_{0.01}$	$1.42_{0.01}$	$0.01_{0.01}$
		Landsat (6)	$1.22_{0.03}$	$1.32_{0.78}$	$1.32_{0.03}$	$1.46_{0.04}$	$0.03_{0.02}$
		LetterRecognition (26)	$2.26_{0.06}$	$2.49_{2.63}$	$2.49_{0.07}$	$2.77_{0.08}$	$0.05_{0.04}$
		MagicGamma (2)	$1.16_{0.01}$	$1.21_{0.49}$	$1.21_{0.01}$	$1.27_{0.02}$	$0.01_{0.01}$
		SensorLessDrive (11)	$0.93_{0.01}$	$0.95_{0.39}$	$0.95_{0.01}$	$0.97_{0.01}$	$0.00_{0.00}$
		Shuttle (7)	$0.89_{0.00}$	$0.90_{0.31}$	$0.90_{0.00}$	$0.91_{0.00}$	$0.00_{0.00}$

mates: Monte Carlo averaging uses both the test and the calibration data, whereas our estimates use only the calibration data. Our estimation procedures provide both point and interval (with $\gamma = 0.1$) estimates; they are obtained from Section 5.1 (with valid confidence intervals) for the l_1 and 0-1 loss non-conformity functions, and from Section 5.2 for the other functions.

Table 3 illustrates the Monte Carlo average and our estimates for the marginal expected prediction set size. The means of our point estimates are close to that of the Monte Carlo average, with the standard deviations being comparable or smaller despite using 3× fewer data points for the approximation. This is also reflected in the low absolute error between our individual point estimates and the mean Monte Carlo average. The only exception is when using the 0-1 loss non-conformity function on Landsat, but the standard deviations of the estimates are high under this setting. Additionally, our interval estimates provide lower-upper bounds on the expected set size in practice (with the bounds being

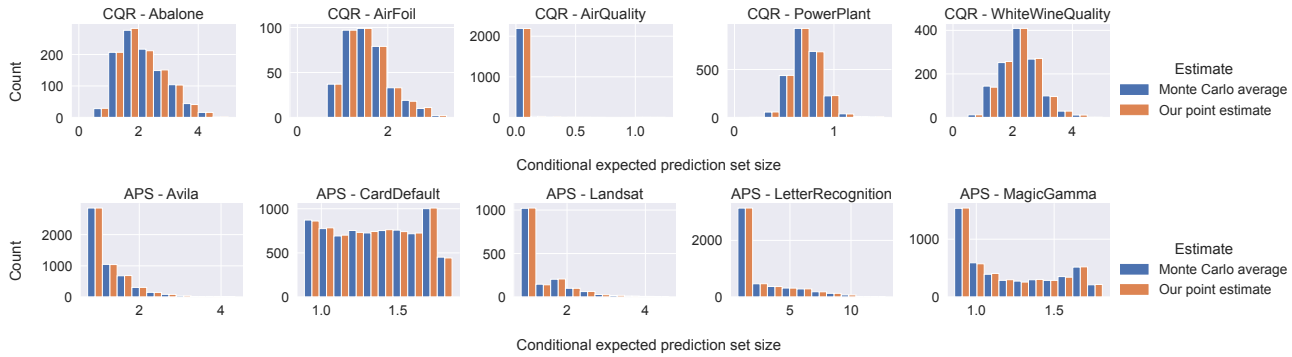

Figure 1: **Expected prediction set sizes conditioned on the test datum feature.** We illustrate the expected sizes of split conformal prediction sets conditioned on varying test datum features using different UCI datasets. We use CQR [Romano et al., 2019] for regression in the top row and APS [Romano et al., 2020] for classification in the bottom row. The estimates are obtained via Monte Carlo averaging and our point estimates (refer to the legend for the color scheme); they are depicted as a histogram with side-by-side bars for comparison.

around our point estimate). These results corroborate the efficacy of our derived practical estimates.

6.3 Conditional Expected Prediction Set Size

Next, we analyze the expected size of prediction sets conditioned on the test inputs. Here, we consider the non-conformity functions CQR [Romano et al., 2019] for regression and APS [Romano et al., 2020] for classification, with other functions included in Appendix C.5.

We follow a similar setup as before, but instead of using 50% of the data as test, we use 25% and fix them across the different data splits to compare the conditional expected prediction set sizes on these particular inputs; we will use the remaining 25% as accessible data points. As before, we compare our derived point estimate with Monte Carlo averaging. In this case, the Monte Carlo average is the average size of the prediction sets constructed using the split conformal algorithm and the calibration data, across all data splits, on a fixed test datum. On the other hand, we obtain our point estimate from Section 5.2 using the accessible data (without having access to the calibration data), and condition on a fixed test datum feature with its feature-specific multiplicative factor (cf. Corollary 2).

Figure 1 depicts histograms of the two estimates for the expected set sizes conditioned on varying test inputs. The plots look identical for the two estimates, despite our point estimate not having seen the calibration data. This further corroborates the efficacy of our estimates.

7 CONCLUSIONS

In this paper, we have studied the expected size of the prediction sets constructed by the split conformal framework. We begin by theoretically quantifying the (marginal and conditional) expected prediction set size (cf. Section 4). Consequently, we derive practical estimation procedures that produce point estimates and high-probability interval bounds for the expected set size (cf. Section 5); these procedures require data to be collected only once to produce reliable estimates. Additionally, we corroborate our results experimentally on real-world regression and classification problems and demonstrate the efficacy of our estimates in practice. Returning to our company and customers in Section 1, the company now has the tools to provide estimates of the expected set size, which allows potential customers to reliably evaluate the company's conformal system.

Acknowledgements

Guneet S. Dhillon is supported by the Clarendon Fund Scholarship, University of Oxford. Tom Rainforth is supported by the UK EPSRC grant EP/Y037200/1.

References

A. N. Angelopoulos and S. Bates. Conformal prediction: A gentle introduction. *Foundations and Trends® in Machine Learning*, 16(4):494–591, 2023. 3

A. N. Angelopoulos, S. Bates, M. Jordan, and J. Malik. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations*, 2021. 1, 3, 19

R. F. Barber, E. J. Candès, A. Ramdas, and R. J. Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023. 16

A. Bellotti. Optimized conformal classification using gradient descent approximation. *arXiv preprint arXiv:2105.11255*, 2021. 3

L. Breiman. Random forests. *Machine Learning*, 45(1): 5–32, 2001. 7

J. Cherian and L. Bronner. How The Washington Post estimates outstanding votes for the 2020 presidential election, 2020. 1

A. Dixit, L. Lindemann, S. X. Wei, M. Cleaveland, G. J. Pappas, and J. W. Burdick. Adaptive conformal prediction for motion planning among dynamic agents. In N. Matni, M. Morari, and G. J. Pappas, editors, *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pages 300–314. PMLR, 2023. 1

A. Dvoretzky, J. Kiefer, and J. Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, 27(3):642–669, 1956. 6, 15, 24

A. Fisch, T. Schuster, T. Jaakkola, and D. Barzilay. Few-shot conformal prediction with auxiliary tasks. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3329–3339. PMLR, 2021a. 1

A. Fisch, T. Schuster, T. S. Jaakkola, and R. Barzilay. Efficient conformal prediction via cascaded inference with expanded admission. In *International Conference on Learning Representations*, 2021b. 1

A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 148–155. Morgan Kaufmann Publishers Inc., 1998. 3

M. Kelly, R. Longjohn, and K. Nottingham. The UCI machine learning repository, 2023. URL `https://archive.ics.uci.edu`. 7, 19, 20

D. Kivaranovic, K. D. Johnson, and H. Leeb. Adaptive, distribution-free prediction intervals for deep networks. In S. Chiappa and R. Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4346–4356. PMLR, 2020. 3

R. Koenker and G. Bassett. Regression quantiles. *Econometrica*, 46(1):33–50, 1978. 18

J. Lei. Classification with confidence. *Biometrika*, 101 (4):755–769, 2014. 1, 3

J. Lei and L. Wasserman. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(1):71–96, 2013. 1, 3

J. Lei, J. Robins, and L. Wasserman. Distribution-free prediction sets. *Journal of the American Statistical Association*, 108(501):278–287, 2013. 1, 3

J. Lei, A. Rinaldo, and L. Wasserman. A conformal prediction approach to explore functional data. *Annals of Mathematics and Artificial Intelligence*, 74 (1):29–43, 2015. 1, 2, 3

L. Lindemann, M. Cleaveland, G. Shim, and G. J. Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 8(8):5116–5123, 2023. 1

C. Lu and J. Kalpathy-Cramer. Distribution-free federated learning with conformal predictions. *arXiv preprint arXiv:2110.07661*, 2021. 1

C. Lu, A. Lemay, K. Chang, K. Höbel, and J. Kalpathy-Cramer. Fair conformal predictors for applications in medical imaging. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12008–12016, 2022. 1

P. Massart. The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality. *The Annals of Probability*, 18 (3):1269–1283, 1990. 6, 15, 24

N. Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7(35):983–999, 2006. 7

H. Papadopoulos, K. Proedrou, V. Vovk, and A. Gammerman. Inductive confidence machines for regression. In T. Elomaa, H. Mannila, and H. Toivonen, editors, *Machine Learning: ECML 2002*, pages 345–356. Springer Berlin Heidelberg, 2002. 1, 2, 3, 16

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 7

T. Rainforth, R. Cornish, H. Yang, A. Warrington, and F. Wood. On nesting Monte Carlo estimators. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4267–4276. PMLR, 2018. 7

Y. Romano, E. Patterson, and E. J. Candès. Conformalized quantile regression. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 3, 4, 5, 7, 8, 9, 18, 21

Y. Romano, M. Sesia, and E. J. Candès. Classification with valid and adaptive coverage. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3581–3591. Curran Associates, Inc., 2020. 3, 4, 5, 7, 8, 9, 19, 21

M. Sadinle, J. Lei, and L. Wasserman. Least ambiguous set-valued classifiers with bounded error levels. *Journal of the American Statistical Association*, 114 (525):223–234, 2019. 1, 3, 4, 5, 7, 8, 17

C. Saunders, A. Gammerman, and V. Vovk. Transduction with confidence and credibility. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 722–726. Morgan Kaufmann Publishers Inc., 1999. 3

T. Schuster, A. Fisch, T. Jaakkola, and R. Barzilay. Consistent accelerated inference via confident adaptive transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4962–4979. Association for Computational Linguistics, 2021. 1

G. Shafer and V. Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(12): 371–421, 2008. 1, 2, 4

D. Stutz, K. D. Dvijotham, A. T. Cemgil, and A. Doucet. Learning optimal conformal classifiers. In *International Conference on Learning Representations*, 2022. 3

R. J. Tibshirani, R. Foygel Barber, E. J. Candès, and A. Ramdas. Conformal prediction under covariate shift. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 7, 16

V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer-Verlag, Berlin, Heidelberg, 2005. 1, 2, 3, 4, 16

V. Vovk, I. Petej, and V. Fedorova. From conformal to probabilistic prediction. In L. Iliadis, I. Maglogiannis, H. Papadopoulos, S. Sioutas, and C. Makris, editors, *Artificial Intelligence Applications and Innovations*, pages 221–230. Springer Berlin Heidelberg, 2014. 1, 3

V. Vovk, V. Fedorova, I. Nouretdinov, and A. Gammerman. Criteria of efficiency for conformal prediction. In A. Gammerman, Z. Luo, J. Vega, and V. Vovk, editors, *Conformal and Probabilistic Prediction with Applications*, pages 23–39. Springer International Publishing, 2016. 1, 3

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes. We include clear descriptions of our developed methods in Section 5.**
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **Yes. We include analyses of our methods and their properties in Section 5.**
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **Yes. Our code is available at https://github.com/Guneet-Dhillon/expected-conformal-prediction-set-size.**

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **Yes. We include statements of the assumptions made for our theoretical results in Sections 4 and 5.**
   (b) Complete proofs of all theoretical results. **Yes. We include proofs in Appendix A.**
   (c) Clear explanations of any assumptions. **Yes. We include clear explanations of our assumptions in Sections 4 and 5.**

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **Yes. We include the URL to our code.**
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Yes. We include the experimental setup in Section 6 (further in Appendices C and D).**
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes. We include the measures/statistics used in Section 6.**
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **Yes. We used a single 2.7 GHz Dual-Core Intel Core i5 processor.**

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. **Yes.**
   (b) The license information of the assets, if applicable. **Yes.**
   (c) New assets either in the supplemental material or as a URL, if applicable. **Yes.**
   (d) Information about consent from data providers/curators. **Not Applicable.**

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **Not Applicable.**

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. **Not Applicable.**
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **Not Applicable.**
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **Not Applicable.**

A PROOFS

A.1 Proof for Theorem 1

Proof. The expected size of prediction sets under the split conformal prediction framework (cf. Equation (2)) is,

$$\begin{aligned}
\mathbb{E}\left[\left|\hat{C}_\alpha^R(X_{n+1}; Z_{1:n})\right|\right] &= \mathbb{E}\left[\left|\{y \in \mathcal{Y} : \tau_\alpha(R_{1:n}) \geq R(X_{n+1}, y)\}\right|\right] \\
&= \mathbb{E}\left[\int_{\mathcal{Y}} \mathbb{1}\left\{\tau_\alpha(R_{1:n}) \geq R(X_{n+1}, y)\right\} dy\right] \\
&= \int_{\mathcal{Y}} \mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq R(X_{n+1}, y)\right\} dy \\
&= \int_{\mathcal{Y}} \int_{\mathcal{R}} \mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq R(X_{n+1}, y) | R(X_{n+1}, y) = r\right\} P_{R(X_{n+1},y)}(dr)\, dy,
\end{aligned}$$

where, for every label $y \in \mathcal{Y}$, we denote $P_{R(X_{n+1},y)}(dr)$ to be the law of the random variable $R(X_{n+1}, y)$, or equivalently, the push-forward of the marginal distribution of X_{n+1} under the mapping $X_{n+1} \mapsto R(X_{n+1}, y)$. In other words, $(y, A) \in \mathcal{Y} \times \mathcal{B}(\mathcal{R}) \mapsto P_{R(X_{n+1},y)}(A)$ defines a transition kernel, where $\mathcal{B}(\mathcal{R})$ denotes the Borel σ-algebra of the space of non-conformity scores $\mathcal{R}$. Continuing from above, we have that,

$$\begin{aligned}
\mathbb{E}\left[\left|\hat{C}_\alpha^R(X_{n+1}; Z_{1:n})\right|\right] &= \int_{\mathcal{Y}} \int_{\mathcal{R}} \mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq R(X_{n+1}, y) | R(X_{n+1}, y) = r\right\} P_{R(X_{n+1},y)}(dr)\, dy \\
&\overset{(i)}{=} \int_{\mathcal{Y}} \int_{\mathcal{R}} \mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq r\right\} P_{R(X_{n+1},y)}(dr)\, dy \\
&= \int_{\mathcal{R}} \mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq r\right\} \int_{\mathcal{Y}} P_{R(X_{n+1},y)}(dr)\, dy \\
&= \int_{\mathcal{R}} \mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq r\right\} \#_R(dr),
\end{aligned}$$

where (i) follows from the test and the calibration non-conformity scores being independent of each other (since test and calibration data are independent of each other, so are their scores). The measure $\#_R$ is defined as,

$$\#_R(A) = \int_{\mathcal{Y}} P_{R(X_{n+1},y)}(A)\, dy,$$

for $A \in \mathcal{B}(\mathcal{R})$. Note that $\#_R(\mathcal{R}) = |\mathcal{Y}|$, which may be infinite, for instance, when $\mathcal{Y} = \mathbb{R}$.

If $\#_R$ is absolutely continuous w.r.t. the reference measure, then $\#_R(dr) = \#_R(r)dr$. If the law of $R(X_{n+1}, y)$ is absolutely continuous w.r.t. the reference measure on $\mathcal{R}$, i.e., $P_{R(X_{n+1},y)}(dr) = p_{R(X_{n+1},y)}(r)dr$, then $\#_R$ is also absolutely continuous w.r.t. the reference measure, with the following density,

$$\#_R(r) = \int_{\mathcal{Y}} p_{R(X_{n+1},y)}(r)\, dy,$$

where we use the same symbol for the density. This is the multiplicative factor, defined in Equation (6).

Continuing from above, we quantify the expected size of the prediction sets as follows,

$$\mathbb{E}\left[\left|\hat{C}_\alpha^R(X_{n+1}; Z_{1:n})\right|\right] = \int_{\mathcal{R}} \mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq r\right\} \#_R(r)dr,$$

which is the desired formulation in Equation (3).

Furthermore, the calibration non-conformity scores are i.i.d. with the probability density/mass function p_R. As a result, the individual (identical) probabilities for each calibration non-conformity score being strictly less than r is $\tilde{P}_R(r)$ (cf. Equation (4)). Additionally, the threshold $\tau_\alpha(R_{1:n})$ is the $\lceil (1-\alpha)(n+1) \rceil$'th smallest value in the augmented set of calibration non-conformity scores $\{R_1, \ldots, R_n, \infty\}$. Then, for the event $\tau_\alpha(R_{1:n}) \geq r$ to occur, at most $n_\alpha = \lceil (1-\alpha)(n+1) \rceil - 1$ of the n calibration non-conformity scores can be strictly smaller than r. If we consider a calibration score being strictly smaller than r as a success, we can simplify the event

$\tau_\alpha(R_{1:n}) \geq r$ to the event $B(n, \tilde{P}_R(r)) \leq n_\alpha$, where $B(n, \tilde{P}_R(r))$ is a binomial random variable with n trials and success probability $\tilde{P}_R(r)$. Finally, the probability of the event $\tau_\alpha(R_{1:n}) \geq r$ simplifies to the following,

$$\mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq r\right\} = \mathbb{P}\left\{B\left(n, \tilde{P}_R\left(r\right)\right) \leq n_\alpha\right\} = P_{B\left(n, \tilde{P}_R(r)\right)}\left(n_\alpha\right).$$

Making the above simplification in Equation (3) leads to the desired expected set size in Equation (5).

□

A.2 Proof for Corollary 2

Proof. The expected size of split conformal prediction sets conditioned on the test datum feature $X_{n+1} = x_{n+1}$ is,

$$\begin{aligned}
\mathbb{E}\left[\left|\hat{C}^R_\alpha(X_{n+1}; Z_{1:n})\right| \middle| X_{n+1} = x_{n+1}\right] &= \mathbb{E}\left[|\{y \in \mathcal{Y} : \tau_\alpha(R_{1:n}) \geq R(X_{n+1}, y)\}| | X_{n+1} = x_{n+1}\right] \\
&= \mathbb{E}\left[\int_{\mathcal{Y}} \mathbb{1}\left\{\tau_\alpha(R_{1:n}) \geq R(X_{n+1}, y)\right\} dy \middle| X_{n+1} = x_{n+1}\right] \\
&= \int_{\mathcal{Y}} \mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq R(X_{n+1}, y) | X_{n+1} = x_{n+1}\right\} dy \\
&\overset{(i)}{=} \int_{\mathcal{Y}} \mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq R(x_{n+1}, y)\right\} dy \\
&= \int_{\mathcal{Y}} \int_{\mathcal{R}} \mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq r\right\} \delta_{R(x_{n+1}, y)}(dr) dy \\
&= \int_{\mathcal{R}} \mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq r\right\} \int_{\mathcal{Y}} \delta_{R(x_{n+1}, y)}(dr) dy \\
&= \int_{\mathcal{R}} \mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq r\right\} \#_R(dr; x_{n+1}),
\end{aligned}$$

where (i) follows from the test and the calibration non-conformity scores being independent of each other (since the test and the calibration data are independent of each other). The measure is $\#_R(dr; x_{n+1}) = \int_{\mathcal{Y}} \delta_{R(x_{n+1}, y)}(dr) dy$, where $\delta_{R(x_{n+1}, y)}$ is the Dirac delta distribution that places all of its probability mass on $R(x_{n+1}, y)$. We define its Radon-Nikodym w.r.t. the reference measure as $\#_R(r; x_{n+1}) = \int_{\mathcal{Y}} \delta_{R(x_{n+1}, y)}(r) dy$, when it exists; this is the feature-specific multiplicative factor (cf. Section 4.2). As a result, we obtain $\#_R(dr; x_{n+1}) = \#_R(r; x_{n+1}) dr$.

Continuing, we quantify the expected prediction set size conditioned on the test datum feature $X_{n+1} = x_{n+1}$ as,

$$\mathbb{E}\left[\left|\hat{C}^R_\alpha(X_{n+1}; Z_{1:n})\right| \middle| X_{n+1} = x_{n+1}\right] = \int_{\mathcal{R}} \mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq r\right\} \#_R(r; x_{n+1}) dr,$$

which is the desired formulation in Equation (7). Furthermore, following the proof in Appendix A.1, we can simplify $\mathbb{P}\{\tau_\alpha(R_{1:n}) \geq r\} = P_{B(n, \tilde{P}_R(r))}(n_\alpha)$, to get the desired conditional expected prediction set size in Equation (8).

□

A.3 Proof for Corollary 3

Proof. For each calibration non-conformity score R_i, for $i = 1, \ldots, n$, we introduce a new random variable $V_i = -R_i$. Since $R_{1:n}$ are i.i.d., the random variables $V_{1:n}$ are i.i.d. as well, and we denote their cumulative distribution function as P_V. We recognize that $\tilde{P}_R$ (cf. Equation (4)) and P_V are related in the following way,

$$\tilde{P}_R\left(r\right) = \mathbb{P}\left\{R_1 < r\right\} = \mathbb{P}\left\{-R_1 > -r\right\} = \mathbb{P}\left\{V_1 > -r\right\} = 1 - \mathbb{P}\left\{V_1 \leq -r\right\} = 1 - P_V\left(-r\right).$$

Equivalently, the empirical approximation $\tilde{P}^{\text{emp}}_R$ (cf. Equation (9)) approximates P_V in the following way,

$$\begin{aligned}
&\tilde{P}^{\text{emp}}_R\left(r\right) = \frac{1}{k}\sum\nolimits_{i=1}^{k} \mathbb{1}\left\{R'_i < r\right\} = \frac{1}{k}\sum\nolimits_{i=1}^{k} \mathbb{1}\left\{-R'_i > -r\right\} \\
&= \frac{1}{k}\sum\nolimits_{i=1}^{k} \mathbb{1}\left\{V'_i > -r\right\} = 1 - \frac{1}{k}\sum\nolimits_{i=1}^{k} \mathbb{1}\left\{V'_i \leq -r\right\} = 1 - P^{\text{emp}}_V\left(-r\right),
\end{aligned}$$

where we define $V_i' = -R_i'$, for $i = 1, \ldots, k$.

The Dvoretzky–Kiefer–Wolfowitz inequality [Dvoretzky et al., 1956; Massart, 1990] bounds the difference between the cumulative distribution function and its empirical approximation, which can be transformed into confidence intervals. We set $\Delta_{k,\gamma} = \sqrt{\ln(2/\gamma)/2k}$. With probability at least $1-\gamma$, for all $r \in \mathcal{R}$,

$$\begin{aligned} & P_V(-r) \in [P_V^{\text{emp}}(-r) - \Delta_{k,\gamma}, P_V^{\text{emp}}(-r) + \Delta_{k,\gamma}] \\ \iff & 1 - P_V(-r) \in [1 - P_V^{\text{emp}}(-r) - \Delta_{k,\gamma}, 1 - P_V^{\text{emp}}(-r) + \Delta_{k,\gamma}] \\ \iff & \tilde{P}_R(r) \in \left[\tilde{P}_R^{\text{emp}}(r) - \Delta_{k,\gamma}, \tilde{P}_R^{\text{emp}}(r) + \Delta_{k,\gamma}\right]. \end{aligned}$$

This implies that with probability at least $1-\gamma$, for all $r \in \mathcal{R}$,

$$P_{B\left(n,\tilde{P}_R(r)\right)}(n_\alpha) \in \left\{P_{B(n,p)}(n_\alpha)\right\}_{p \in \left[\tilde{P}_R^{\text{emp}}(r) - \Delta_{k,\gamma}, \tilde{P}_R^{\text{emp}}(r) + \Delta_{k,\gamma}\right]}.$$

Since $P_{B(n,p)}(n_\alpha)$ is a non-increasing function in p, with probability at least $1-\gamma$, for all $r \in \mathcal{R}$,

$$P_{B\left(n,\tilde{P}_R(r)\right)}(n_\alpha) \in \left[P_{B\left(n,\tilde{P}_R^{\text{emp}}(r) + \Delta_{k,\gamma}\right)}(n_\alpha), P_{B\left(n,\tilde{P}_R^{\text{emp}}(r) - \Delta_{k,\gamma}\right)}(n_\alpha)\right].$$

Since this holds for all $r \in \mathcal{R}$ simultaneously, with probability at least $1-\gamma$,

$$\mathbb{E}\left[\left|\hat{C}_\alpha^R(X_{n+1}; Z_{1:n})\right|\right] \overset{(i)}{=} \int_{\mathcal{R}} P_{B\left(n,\tilde{P}_R(r)\right)}(n_\alpha) \#_R(r) dr \in$$

$$\left[\int_{\mathcal{R}} P_{B\left(n,\tilde{P}_R^{\text{emp}}(r) + \Delta_{k,\gamma}\right)}(n_\alpha) \#_R(r) dr, \int_{\mathcal{R}} P_{B\left(n,\tilde{P}_R^{\text{emp}}(r) - \Delta_{k,\gamma}\right)}(n_\alpha) \#_R(r) dr\right],$$

where (i) follows from Equation (5) and the multiplicative factor $\#_R(r)$ is known. This is the desired confidence interval for the expected size of the prediction sets in Equation (10) of Corollary 3.

□

A.4 Corollary 4

In addition to Corollary 2 where we quantify the expected prediction set size conditioned on a test datum feature, we can condition on the calibration data $Z_{1:n} = z_{1:n}$ instead. We summarize the corresponding result below.

Corollary 4 (Expected size of prediction sets conditioned on the calibration data)**.** *If the test and the calibration non-conformity scores are independent of each other, then the expected size of the split conformal prediction sets conditioned on the calibration data $Z_{1:n} = z_{1:n}$ (and $r_i = R(z_i)$, for $i = 1, \ldots, n$) is given by the following,*

$$\mathbb{E}\left[\left|\hat{C}_\alpha^R(X_{n+1}; Z_{1:n})\right| \middle| Z_{1:n} = z_{1:n}\right] = \int_{\mathcal{R}} \mathbb{1}\left\{\tau_\alpha(r_{1:n}) \geq r\right\} \#_R(r) dr. \tag{12}$$

Proof. The expected size of split conformal prediction sets conditioned on the calibration data $Z_{1:n} = z_{1:n}$ is,

$$\begin{aligned} \mathbb{E}\left[\left|\hat{C}_\alpha^R(X_{n+1}; Z_{1:n})\right| \middle| Z_{1:n} = z_{1:n}\right] &= \mathbb{E}\left[|\{y \in \mathcal{Y} : \tau_\alpha(R_{1:n}) \geq R(X_{n+1}, y)\}| | Z_{1:n} = z_{1:n}\right] \\ &= \mathbb{E}\left[\int_{\mathcal{Y}} \mathbb{1}\left\{\tau_\alpha(R_{1:n}) \geq R(X_{n+1}, y)\right\} dy \middle| Z_{1:n} = z_{1:n}\right] \\ &= \int_{\mathcal{Y}} \mathbb{P}\left\{\tau_\alpha(R_{1:n}) \geq R(X_{n+1}, y) | Z_{1:n} = z_{1:n}\right\} dy \\ &\overset{(i)}{=} \int_{\mathcal{Y}} \mathbb{P}\left\{\tau_\alpha(r_{1:n}) \geq R(X_{n+1}, y)\right\} dy \\ &\overset{(ii)}{=} \int_{\mathcal{R}} \mathbb{1}\left\{\tau_\alpha(r_{1:n}) \geq r\right\} \#_R(r) dr, \end{aligned}$$

where (i) follows from the test and the calibration non-conformity scores being independent of each other (since the test and the calibration data are independent of each other), and (ii) follows from the proof in Appendix A.1.

□

A.5 Proof for Practical Estimation (Unknown Multiplicative Factor)

Under the setting where the multiplicative factor is intractable due to its dependence on the data distribution and/or the machine learning model, we can re-arrange the formulation of the expected prediction set size in Equation (5) to get rid of this factor. Using Equations (5) and (6), we obtain the following,

$$\mathbb{E}\left[\left|\hat{C}_\alpha^R(X_{n+1}; Z_{1:n})\right|\right] = \int_{\mathcal{R}} P_{B\left(n,\tilde{P}_R(r)\right)}(n_\alpha)\, \#_R(r) dr = \int_{\mathcal{R}} P_{B\left(n,\tilde{P}_R(r)\right)}(n_\alpha) \int_{\mathcal{Y}} p_{R(X_{n+1},y)}(r)\, dy\, dr$$
$$= \int_{\mathcal{Y}} \int_{\mathcal{R}} P_{B\left(n,\tilde{P}_R(r)\right)}(n_\alpha)\, p_{R(X_{n+1},y)}(r)\, dr\, dy = \int_{\mathcal{Y}} \mathbb{E}\left[P_{B\left(n,\tilde{P}_R(R(X_{n+1},y))\right)}(n_\alpha)\right] dy,$$

where the expectation term is evaluated over the random variable $R(X_{n+1}, y)$. This derives Equation (11).

B MULTIPLICATIVE FACTOR

The multiplicative factor (cf. Equation (6)) is responsible for translating the reference measure on the space of non-conformity scores to the reference measure on the label space. Here we provide the derivation of this factor under different settings used in Section 6 for the experiments. Note that we overload $\int_{\mathcal{Y}} dy$ to be the Lebesgue measure when $\mathcal{Y}$ is continuous and the counting measure when it is discrete (amounting to a sum over $\mathcal{Y}$).

B.1 l_p Loss for Regression

We begin with regression problems, where the label space is the set of reals, i.e., $\mathcal{Y} = \mathbb{R}$. A common non-conformity function for such problems involves a machine learning model M and the l_1 loss [Papadopoulos et al., 2002; Vovk et al., 2005; Tibshirani et al., 2019; Barber et al., 2023]. We generalize this to any l_p loss, with $p \geq 1$. Then, the non-conformity function is given by $R(x, y) = |M(x) - y|^p$ and the space of non-conformity scores is $\mathcal{R} = [0, \infty)$.

We can define the cumulative distribution function of $R(X_{n+1}, y)$ as follows,

$$P_{R(X_{n+1},y)}(r) = \mathbb{P}\left\{R(X_{n+1}, y) \leq r\right\} = \mathbb{P}\left\{|M(X_{n+1}) - y|^p \leq r\right\} = \mathbb{P}\left\{|M(X_{n+1}) - y| \leq r^{1/p}\right\}$$
$$= \mathbb{P}\left\{y - r^{1/p} \leq M(X_{n+1}) \leq y + r^{1/p}\right\} = \mathbb{P}\left\{M(X_{n+1}) \leq y + r^{1/p}\right\} - \mathbb{P}\left\{M(X_{n+1}) < y - r^{1/p}\right\}$$
$$\overset{(i)}{=} \mathbb{P}\left\{M(X_{n+1}) \leq y + r^{1/p}\right\} - \mathbb{P}\left\{M(X_{n+1}) \leq y - r^{1/p}\right\} = P_{M(X_{n+1})}\left(y + r^{1/p}\right) - P_{M(X_{n+1})}\left(y - r^{1/p}\right),$$

where (i) follows from the prediction $M(X_{n+1})$ being continuous. Further, differentiating the above with respect to r, we get the probability density function of $R(X_{n+1}, y)$ as follows,

$$p_{R(X_{n+1},y)}(r) = \frac{r^{1/p-1}}{p} p_{M(X_{n+1})}\left(y + r^{1/p}\right) + \frac{r^{1/p-1}}{p} p_{M(X_{n+1})}\left(y - r^{1/p}\right).$$

Therefore, the multiplicative factor in this setting is given by,

$$\#_R(r) = \int_{\mathbb{R}} p_{R(X_{n+1},y)}(r)\, dy = \int_{\mathbb{R}} \frac{r^{1/p-1}}{p} p_{M(X_{n+1})}\left(y + r^{1/p}\right) dy + \int_{\mathbb{R}} \frac{r^{1/p-1}}{p} p_{M(X_{n+1})}\left(y - r^{1/p}\right) dy$$
$$= \frac{r^{1/p-1}}{p}\left(\int_{\mathbb{R}} p_{M(X_{n+1})}\left(y + r^{1/p}\right) dy + \int_{\mathbb{R}} p_{M(X_{n+1})}\left(y - r^{1/p}\right) dy\right)$$
$$\overset{(ii)}{=} \frac{r^{1/p-1}}{p}\left(\int_{\mathbb{R}} p_{M(X_{n+1})}(u)\, du + \int_{\mathbb{R}} p_{M(X_{n+1})}(v)\, dv\right) = \frac{r^{1/p-1}}{p}(1+1) = \frac{2r^{1/p-1}}{p},$$

where (ii) follows from a change of variables with $u = y + r^{1/p}$ and $v = y - r^{1/p}$.

l_p Loss for High-Dimensional Regression We do not restrict ourselves to one-dimensional regression; we can further generalize the above to m-dimensional regression problems, where the label space is $\mathcal{Y} = \mathbb{R}^m$, with $m \geq 1$. The non-conformity function involves a machine learning model M and the l_p loss, with $p \geq 1$. Then, the non-conformity function is given by $R(x, y) = \|M(x) - y\|_p^p$ and the space of non-conformity scores is $\mathcal{R} = [0, \infty)$.

The multiplicative factor in this setting is given by,

$$\begin{aligned}\#_R(r) &= \int_{\mathbb{R}^m} p_{R(X_{n+1},y)}(r)\,dy = \int_{\mathbb{R}^m} \frac{d}{dr} P_{R(X_{n+1},y)}(r)\,dy = \frac{d}{dr}\int_{\mathbb{R}^m} P_{R(X_{n+1},y)}(r)\,dy \\ &= \frac{d}{dr}\int_{\mathbb{R}^m} \mathbb{P}\left\{R(X_{n+1},y)\le r\right\} dy = \frac{d}{dr}\int_{\mathbb{R}^m} \mathbb{P}\left\{\|M(X_{n+1})-y\|_p^p \le r\right\} dy \\ &= \frac{d}{dr}\int_{\mathbb{R}^m} \mathbb{P}\left\{\|M(X_{n+1})-y\|_p \le r^{1/p}\right\} dy.\end{aligned}$$

We denote $B_m^p(c,r)$ to be a m-dimensional l_p-ball with center c and radius r, and $V_m^p(r)$ to be the volume of a m-dimensional l_p-ball with radius r. Continuing from above, we have that,

$$\begin{aligned}\#_R(r) &= \frac{d}{dr}\int_{\mathbb{R}^m} \mathbb{P}\left\{\|M(X_{n+1})-y\|_p \le r^{1/p}\right\} dy = \frac{d}{dr}\int_{\mathbb{R}^m} \mathbb{P}\left\{y\in B_m^p\left(M(X_{n+1}), r^{1/p}\right)\right\} dy \\ &= \frac{d}{dr}\int_{\mathbb{R}^m}\int_{\mathbb{R}^m} \mathbb{P}\left\{y\in B_m^p\left(M(X_{n+1}), r^{1/p}\right)\middle| M(X_{n+1})=c\right\} p_{M(X_{n+1})}(c)\,dc\,dy \\ &= \frac{d}{dr}\int_{\mathbb{R}^m}\int_{\mathbb{R}^m} \mathbb{1}\left\{y\in B_m^p\left(c, r^{1/p}\right)\right\} p_{M(X_{n+1})}(c)\,dc\,dy = \frac{d}{dr}\int_{\mathbb{R}^m} p_{M(X_{n+1})}(c)\int_{\mathbb{R}^m} \mathbb{1}\left\{y\in B_m^p\left(c, r^{1/p}\right)\right\} dy\,dc \\ &= \frac{d}{dr}\int_{\mathbb{R}^m} p_{M(X_{n+1})}(c)\,V_m^p\left(r^{1/p}\right) dc = \frac{d}{dr} V_m^p\left(r^{1/p}\right)\int_{\mathbb{R}^m} p_{M(X_{n+1})}(c)\,dc = \frac{d}{dr} V_m^p\left(r^{1/p}\right).\end{aligned}$$

Furthermore, the volume of a m-dimensional l_p-ball with radius r is $V_m^p(r) = (2\Gamma(1/p+1))^m r^m/\Gamma(m/p+1)$. Therefore, the multiplicative factor in this setting is given by,

$$\#_R(r) = \frac{d}{dr} V_m^p\left(r^{1/p}\right) = \frac{d}{dr}\frac{(2\Gamma(1/p+1))^m}{\Gamma(m/p+1)} r^{m/p} = \frac{(2\Gamma(1/p+1))^m}{\Gamma(m/p+1)}\frac{m}{p} r^{m/p-1}.$$

Note that this is a generalization of our previous result for 1-dimensional regression ($\mathcal{Y}=\mathbb{R}$) to higher dimensions ($\mathcal{Y}=\mathbb{R}^m$). Indeed, by substituting $m=1$, we recover the multiplicative factor $\#_R(r) = 2r^{1/p-1}/p$ as before.

B.2 0-1 Loss for Classification

Here we consider classification problems, where the label space is discrete. The machine learning model M predicts a label directly, i.e., $M(x)\in\mathcal{Y}$ for an input feature $x\in\mathcal{X}$. We consider the 0-1 loss which takes the value 0 if the prediction is correct and 1 when incorrect. Then, the non-conformity function is given by $R(x,y) = \mathbb{1}\{M(x)\neq y\}$ and the space of non-conformity scores is $\mathcal{R}=\{0,1\}$. The multiplicative factor in this setting is given by,

$$\begin{aligned}\#_R(r) &= \sum_{y\in\mathcal{Y}} p_{R(X_{n+1},y)}(r) = \sum_{y\in\mathcal{Y}} \mathbb{P}\left\{R(X_{n+1},y)=r\right\} = \sum_{y\in\mathcal{Y}} \mathbb{P}\left\{\mathbb{1}\left\{M(x)\neq y\right\}=r\right\} \\ &= \begin{cases}\sum_{y\in\mathcal{Y}} \mathbb{P}\left\{M(x)=y\right\}, & r=0\\ \sum_{y\in\mathcal{Y}} \mathbb{P}\left\{M(x)\neq y\right\}, & r=1\end{cases} = \begin{cases}\sum_{y\in\mathcal{Y}} \mathbb{P}\left\{M(x)=y\right\}, & r=0\\ \sum_{y\in\mathcal{Y}} \left(1-\mathbb{P}\left\{M(x)=y\right\}\right), & r=1\end{cases} = \begin{cases}1, & r=0\\ |\mathcal{Y}|-1, & r=1\end{cases}.\end{aligned}$$

B.3 Other Settings

There are many other non-conformity functions proposed for regression and classification problems. However, in some cases, the multiplicative factor can depend on the distribution of data and the machine learning model used.

Least Ambiguous Set-Valued Classifiers (LAC) Sadinle et al. [2019] propose LAC that provably construct prediction sets with minimum expected size if the predicted probabilities are correct; this does not hold in practice, but the predicted sets are small. In this case, the machine learning model M predicts a probability distribution over the labels; we denote $M_y(x)$ as the predicted probability for label $y\in\mathcal{Y}$ for an input feature $x\in\mathcal{X}$. The non-conformity function is given by $R(x,y) = 1 - M_y(x)$ and the space of non-conformity scores is $\mathcal{R}=[0,1]$.

We can define the cumulative distribution function of $R(X_{n+1},y)$ as follows,

$$\begin{aligned}&P_{R(X_{n+1},y)}(r) = \mathbb{P}\left\{R(X_{n+1},y)\le r\right\} = \mathbb{P}\left\{1-M_y(X_{n+1})\le r\right\} = \mathbb{P}\left\{M_y(X_{n+1})\ge 1-r\right\} \\ &= 1-\mathbb{P}\left\{M_y(X_{n+1})<1-r\right\} \overset{(i)}{=} 1-\mathbb{P}\left\{M_y(X_{n+1})\le 1-r\right\} = 1-P_{M_y(X_{n+1})}(1-r),\end{aligned}$$

where (i) follows from the prediction $M_y(X_{n+1})$ being continuous. Further, differentiating the above with respect to r, we get the probability density function of $R(X_{n+1}, y)$ as follows,

$$p_{R(X_{n+1},y)}(r) = p_{M_y(X_{n+1})}(1-r).$$

Therefore, the multiplicative factor in this setting is given by,

$$\#_R(r) = \sum_{y\in\mathcal{Y}} p_{R(X_{n+1},y)}(r) = \sum_{y\in\mathcal{Y}} p_{M_y(X_{n+1})}(1-r),$$

which is dependent on the data distribution and the machine learning model. Consequently, the multiplicative factor cannot be analytically solved under this setting without making any further assumptions.

Conformalized Quantile Regression (CQR) Romano et al. [2019] propose a non-conformity function for regression ($\mathcal{Y} = \mathbb{R}$). In this case, the machine learning model is trained using quantile regression [Koenker and Bassett, 1978] to have two outputs $M_{\alpha/2}(x), M_{1-\alpha/2}(x) \in \mathbb{R}$, corresponding to predictions of the $(\alpha/2)$'th and $(1-\alpha/2)$'th level quantiles respectively, conditioned on an input feature $x \in \mathcal{X}$. Further, the proposed non-conformity function is a loss on the predicted quantile interval, given by $R(x,y) = \max\{M_{\alpha/2}(x) - y, y - M_{1-\alpha/2}(x)\}$. For ease of notation when deriving the associated multiplicative factor, we set $M(x) = (M_{1-\alpha/2}(x) + M_{\alpha/2}(x))/2 \in \mathbb{R}$ and $M_\Delta(x) = (M_{1-\alpha/2}(x) - M_{\alpha/2}(x))/2 \in \mathbb{R}_{\geq 0}$. Then, the non-conformity function can be rewritten as $R(x,y) = \max\{M(x) - M_\Delta(x) - y, y - M(x) - M_\Delta(x)\}$ and the space of non-conformity scores is $\mathcal{R} = \mathbb{R}$.

The multiplicative factor in this setting is given by,

$$\begin{aligned}
\#_R(r) &= \int_{\mathbb{R}} p_{R(X_{n+1},y)}(r)\,dy = \int_{\mathbb{R}} \frac{d}{dr} P_{R(X_{n+1},y)}(r)\,dy = \frac{d}{dr}\int_{\mathbb{R}} P_{R(X_{n+1},y)}(r)\,dy = \frac{d}{dr}\int_{\mathbb{R}} \mathbb{P}\{R(X_{n+1},y)\le r\}\,dy \\
&= \frac{d}{dr}\int_{\mathbb{R}} \mathbb{P}\{\max\{M(X_{n+1}) - M_\Delta(X_{n+1}) - y, y - M(X_{n+1}) - M_\Delta(X_{n+1})\} \le r\}\,dy \\
&= \frac{d}{dr}\int_{\mathbb{R}} \mathbb{P}\{y \in [M(X_{n+1}) - M_\Delta(X_{n+1}) - r, M(X_{n+1}) + M_\Delta(X_{n+1}) + r]\}\,dy \\
&= \frac{d}{dr}\int_{\mathbb{R}}\int_{\mathbb{R}_{\geq 0}}\int_{\mathbb{R}} \mathbb{P}\{y\in[M(X_{n+1}) - M_\Delta(X_{n+1}) - r, M(X_{n+1}) + M_\Delta(X_{n+1}) + r] | M(X_{n+1})=c, M_\Delta(X_{n+1})=\delta\} \\
&\qquad\qquad p_{M(X_{n+1}),M_\Delta(X_{n+1})}(c,\delta)\,dc\,d\delta\,dy \\
&= \frac{d}{dr}\int_{\mathbb{R}}\int_{\mathbb{R}_{\geq 0}}\int_{\mathbb{R}} \mathbb{1}\{y\in[c-\delta-r, c+\delta+r]\}\, p_{M(X_{n+1}),M_\Delta(X_{n+1})}(c,\delta)\,dc\,d\delta\,dy \\
&= \frac{d}{dr}\int_{\mathbb{R}_{\geq 0}}\int_{\mathbb{R}} p_{M(X_{n+1}),M_\Delta(X_{n+1})}(c,\delta)\int_{\mathbb{R}} \mathbb{1}\{y\in[c-\delta-r, c+\delta+r]\}\,dy\,dc\,d\delta \\
&= \frac{d}{dr}\int_{\mathbb{R}_{\geq 0}}\int_{\mathbb{R}} p_{M(X_{n+1}),M_\Delta(X_{n+1})}(c,\delta)\,2(\delta+r)\mathbb{1}\{\delta+r\ge 0\}\,dc\,d\delta \\
&= \frac{d}{dr}\int_{\mathbb{R}_{\geq 0}} 2(\delta+r)\mathbb{1}\{\delta+r\ge 0\}\int_{\mathbb{R}} p_{M(X_{n+1}),M_\Delta(X_{n+1})}(c,\delta)\,dc\,d\delta \\
&= \frac{d}{dr}\int_{\mathbb{R}_{\geq 0}} 2(\delta+r)\mathbb{1}\{\delta+r\ge 0\}\, p_{M_\Delta(X_{n+1})}(\delta)\,d\delta = \frac{d}{dr}\int_{\max\{0,-r\}}^{\infty} 2(\delta+r) p_{M_\Delta(X_{n+1})}(\delta)\,d\delta \\
&= 2\int_{\max\{0,-r\}}^{\infty} \frac{d}{dr}(\delta+r) p_{M_\Delta(X_{n+1})}(\delta)\,d\delta = 2\int_{\max\{0,-r\}}^{\infty} p_{M_\Delta(X_{n+1})}(\delta)\,d\delta = 2\mathbb{P}\{M_\Delta(X_{n+1})\ge\max\{0,-r\}\} \\
&= 2\left(1-\mathbb{P}\{M_\Delta(X_{n+1}) < \max\{0,-r\}\}\right) \overset{(i)}{=} 2\left(1-\mathbb{P}\{M_\Delta(X_{n+1})\le\max\{0,-r\}\}\right) \\
&= 2\left(1-P_{M_\Delta(X_{n+1})}(\max\{0,-r\})\right) = \begin{cases} 2\left(1-P_{M_\Delta(X_{n+1})}(0)\right), & r\ge 0 \\ 2\left(1-P_{M_\Delta(X_{n+1})}(-r)\right), & r<0 \end{cases} = \begin{cases} 2, & r\ge 0 \\ 2\left(1-P_{M_\Delta(X_{n+1})}(-r)\right), & r<0 \end{cases},
\end{aligned}$$

where (i) follows from $M_\Delta(X_{n+1})$ being continuous. The multiplicative factor is again dependent on the data distribution and the machine learning model, and is therefore intractable without making further assumptions.

Table 4: **Dataset statistics summaries.** We summarize the statistics of the UCI datasets used in our experiments. This includes the number of data points and features for each dataset. For regression datasets, we also include the range of label values; for classification, we also include the number of classes/labels.

Regression ($\mathcal{Y} = \mathbb{R}$)				Classification (discrete $\mathcal{Y}$)			
Dataset	**Number of data points**	**Number of features**	**Range of labels**	**Dataset**	**Number of data points**	**Number of features**	**Number of labels**
Abalone	4177	10	[-2.79, 5.94]	APSFailure	120000	341	2
AirFoil	1503	5	[-3.11, 2.34]	Adult	48842	108	2
AirQuality	8991	14	[-1.35, 7.20]	Avila	20867	10	12
BlogFeedback	60021	280	[-0.72, 3.20]	BankMarketing	41188	63	2
CTSlices	53500	384	[-2.00, 2.25]	CardDefault	30000	23	2
FacebookComments	209074	53	[-0.21, 70.12]	Landsat	6435	36	6
OnlineNews	39644	58	[-8.03, 6.63]	LetterRecognition	20000	16	26
PowerPlant	9568	4	[-2.00, 2.43]	MagicGamma	19020	10	2
Superconductivity	21263	81	[-1.00, 4.39]	SensorLessDrive	58509	48	11
WhiteWineQuality	4898	11	[-3.32, 3.57]	Shuttle	58000	9	7

Table 5: **Prediction error frequencies.** We report the empirically achieved prediction error frequencies for the split conformal prediction framework using different non-conformity functions and UCI datasets (with $\alpha = 0.1$).

Regression ($\mathcal{Y} = \mathbb{R}$)			Classification (discrete $\mathcal{Y}$)			
	Prediction error frequency			**Prediction error frequency**		
Dataset	l_1	**CQR**	**Dataset**	**0-1**	**LAC**	**APS**
Abalone	0.0987	0.0846	APSFailure (2)	0.0055	0.0742	0.0999
AirFoil	0.0987	0.0983	Adult (2)	0.0000	0.0982	0.1001
AirQuality	0.0987	0.0201	Avila (12)	0.0489	0.0973	0.0998
BlogFeedback	0.1001	0.0514	BankMarketing (2)	0.0904	0.0982	0.0999
CTSlices	0.1000	0.0999	CardDefault (2)	0.0000	0.0984	0.0999
FacebookComments	0.1000	0.0449	Landsat (6)	0.0255	0.0982	0.1001
OnlineNews	0.1000	0.0999	LetterRecognition (26)	0.0781	0.0972	0.0999
PowerPlant	0.0997	0.0998	MagicGamma (2)	0.0000	0.0983	0.0999
Superconductivity	0.0999	0.0996	SensorLessDrive (11)	0.0036	0.0938	0.0999
WhiteWineQuality	0.0977	0.0387	Shuttle (7)	0.0007	0.0144	0.0999

Adaptive Prediction Sets (APS) and Regularized Adaptive Prediction Sets (RAPS) Romano et al. [2020]; Angelopoulos et al. [2021] propose non-conformity functions for classification. Romano et al. [2020] propose adaptive prediction sets (APS) that sum the predicted label probabilities in descending order until the label assigned to the data point is included; the corresponding non-conformity function is given by $R(x, y) = UM_y(x) + \sum_{y' \in \mathcal{Y}} \mathbb{1}\{M_{y'}(x) > M_y(x)\} M_{y'}(x)$, where $M_y(x)$ is the predicted probability for label $y \in \mathcal{Y}$ for an input feature $x \in \mathcal{X}$ and $U \sim \mathcal{U}(0, 1)$ is a uniform random variable over $[0, 1]$. Additionally, Angelopoulos et al. [2021] propose regularized adaptive prediction sets (RAPS) that further add a regularization term to penalize the number of labels included in the prediction set. Both these non-conformity functions construct small prediction sets, but their associated multiplicative factors are intractable without making further assumptions.

C EXPERIMENTS ON UCI DATASETS

We illustrate the efficacy of our results experimentally by applying our estimation procedures derived in Section 5 on real-world datasets from the UCI database [Kelly et al., 2023].[4] We summarize the dataset statistics in Table 4.

C.1 Prediction Errors

We include the empirically achieved prediction error frequencies for our implementation of the split conformal prediction framework using different non-conformity functions. This facilitates the evaluation of our implementation in satisfying the requirement in Equation (1). We use the same setup as the one highlighted in Section 6.1.

With the significance level α set to 0.1, the results are illustrated in Table 5. We observe that the error frequencies are either close to or less than the desired bound of $\alpha = 0.1$ for every non-conformity function and dataset.

[4]We use the python package https://github.com/isacarnekvist/ucimlr to access the datasets.

Table 6: **Interval error frequencies.** We report the error frequencies of our individual interval estimates (with $\gamma = 0.1$) bounding the mean Monte Carlo average for different non-conformity functions and UCI datasets.

Regression ($\mathcal{Y} = \mathbb{R}$)			**Classification (discrete $\mathcal{Y}$)**			
	Interval error frequency			**Interval error frequency**		
Dataset	l_1	**CQR**	**Dataset**	**0-1**	**LAC**	**APS**
Abalone	0.0000	0.0000	APSFailure (2)	0.0000	0.4420	0.0040
AirFoil	0.0000	0.0000	Adult (2)	0.0000	0.0000	0.0000
AirQuality	0.0020	1.0000	Avila (12)	0.0000	0.0000	0.0080
BlogFeedback	0.0000	1.0000	BankMarketing (2)	0.0370	0.0000	0.0020
CTSlices	0.0050	0.8820	CardDefault (2)	0.0000	0.0000	0.0000
FacebookComments	0.0000	1.0000	Landsat (6)	0.0000	0.0000	0.0020
OnlineNews	0.0000	0.0000	LetterRecognition (26)	0.0000	0.0000	0.0000
PowerPlant	0.0000	0.0000	MagicGamma (2)	0.0000	0.0000	0.0010
Superconductivity	0.0000	0.0330	SensorLessDrive (11)	0.0000	0.0000	0.0000
WhiteWineQuality	0.0000	0.8920	Shuttle (7)	0.0000	1.0000	0.0050

Table 7: **Marginal expected prediction set sizes (high-dimensional regression).** We illustrate the marginal expected prediction set sizes. The estimates are obtained via Monte Carlo averaging, our point estimates, and our interval estimates (lower-upper bounds with $\gamma = 0.1$). We also compute the absolute errors between our individual point estimates and the mean Monte Carlo average, and the frequencies of error of our individual interval estimates bounding the mean Monte Carlo average. We report the means and standard deviations.

		Marginal expected prediction set size					
	Dataset	**Our interval lower bound**	**Monte Carlo average**	**Our point estimate**	**Our interval upper bound**	**Absolute error**	**Interval error frequency**
$\boldsymbol{l_1}$	Parkinson	$1.33_{0.20}$	$1.85_{0.29}$	$1.85_{0.28}$	$2.71_{0.43}$	$0.24_{0.14}$	0.0010
$\boldsymbol{l_2}$		$1.07_{0.16}$	$1.48_{0.23}$	$1.48_{0.22}$	$2.16_{0.34}$	$0.19_{0.11}$	0.0000

C.2 Interval Estimate Errors

Our estimation procedures in Section 5 provide point and interval estimates for the expected prediction set size. The latter are high-probability bounds, which in fact are valid confidence intervals when the multiplicative factor is known (cf. Corollary 3). We want to validate our results experimentally; however, it is impossible to do so as the true expected set size is unknown in practice. As a proxy, we use the mean Monte Carlo average instead and test its inclusion in our individual interval bounds. With that, we expand on our experimental results in Section 6.2. Note that our interval estimates are obtained from Section 5.1 (with valid confidence intervals) for the l_1 and 0-1 loss non-conformity functions, and from Section 5.2 for the other non-conformity functions.

Table 6 illustrates the frequency of error of our individual estimated intervals (with γ set to 0.1) bounding the mean Monte Carlo average; we would expect these values to be below $\gamma = 0.1$. When the intervals are valid confidence intervals (under the l_1 and 0-1 loss non-conformity functions), the error frequencies are always below 0.1, corroborating our result in Corollary 3. When the intervals are not necessarily valid confidence intervals, they still achieve errors lower than 0.1 on 23/30 instances. In the 7 remaining instances, our point and interval estimates are close to the mean Monte Carlo average, but the standard deviation in the Monte Carlo average estimate itself is high. Note that this is a proxy to the true interval error, which cannot be computed in practice.

C.3 High-Dimensional Regression

Here we consider the Parkinson dataset from the UCI database [Kelly et al., 2023], a 2-dimensional regression dataset with 5875 data points and 19-dimensional features. We use the l_1 and the l_2 loss non-conformity functions with multiplicative factors $\#_R(r) = 4r$ and $\#_R(r) = \pi$ respectively (cf. Appendix B.1). As the multiplicative factors are known, we compute our empirical estimates from Section 5.1. We use the same setup as in Section 6.1.

Table 7 illustrates the Monte Carlo average and our estimates for the marginal expected prediction set size on the Parkinson dataset. We observe trends similar to those in Table 3; the means of our point estimates are close to that of the Monte Carlo average, with the standard deviations being comparable despite using 3× fewer data points. This is also reflected in the low absolute error between our individual point estimates and the mean Monte Carlo average. Additionally, our interval estimates provide lower-upper bounds on the expected set size. Similar to Table 6, the error frequencies of our individual estimated intervals bounding the mean Monte Carlo average are below $\gamma = 0.1$. These results corroborate the efficacy of our estimates on high-dimensional regression problems.

Table 8: **Marginal expected prediction set sizes (insights ablation).** We illustrate changes in the marginal expected prediction set sizes (the Monte Carlo averages) using different non-conformity functions and UCI datasets. The first column corresponds to no change from the setup in Section 6.1. The second corresponds to an increase in the amount of training data. The third corresponds to a decrease in the significance level. The fourth corresponds to an increase in the amount of calibration data. We report the means and standard deviations.

			Marginal expected prediction set size (Monte Carlo average)			
		Dataset	No change	Increase in training data	Decrease in significance level	Increase in calibration data
Regression ($\mathcal{Y} = \mathbb{R}$)	l_1	Abalone	$2.19_{0.09}$	$2.14_{0.09}$	$4.86_{0.36}$	$2.18_{0.06}$
		AirFoil	$1.39_{0.10}$	$1.04_{0.08}$	$2.80_{0.39}$	$1.38_{0.08}$
		AirQuality	$0.02_{0.00}$	$0.01_{0.00}$	$0.07_{0.02}$	$0.02_{0.00}$
		PowerPlant	$0.70_{0.02}$	$0.64_{0.01}$	$1.32_{0.06}$	$0.70_{0.01}$
		WhiteWineQuality	$2.58_{0.08}$	$2.43_{0.08}$	$4.81_{0.27}$	$2.57_{0.05}$
	CQR	Abalone	$2.17_{0.98}$	$1.96_{0.97}$	$4.09_{1.15}$	$2.19_{0.97}$
		AirFoil	$1.58_{0.64}$	$1.22_{0.56}$	$3.01_{0.81}$	$1.57_{0.63}$
		AirQuality	$0.02_{0.11}$	$0.01_{0.06}$	$0.05_{0.19}$	$0.02_{0.11}$
		PowerPlant	$0.73_{0.26}$	$0.68_{0.25}$	$1.27_{0.32}$	$0.73_{0.26}$
		WhiteWineQuality	$2.24_{0.89}$	$2.11_{0.93}$	$5.08_{1.18}$	$2.24_{0.89}$
Classification (discrete $\mathcal{Y}$)	0-1	Avila (12)	$1.00_{0.00}$	$1.00_{0.00}$	$12.00_{0.00}$	$1.00_{0.00}$
		CardDefault (2)	$2.00_{0.00}$	$2.00_{0.00}$	$2.00_{0.00}$	$2.00_{0.00}$
		Landsat (6)	$4.79_{2.14}$	$1.35_{1.28}$	$6.00_{0.00}$	$5.25_{1.78}$
		LetterRecognition (26)	$1.00_{0.00}$	$1.00_{0.00}$	$26.00_{0.00}$	$1.00_{0.00}$
		MagicGamma (2)	$2.00_{0.00}$	$2.00_{0.00}$	$2.00_{0.00}$	$2.00_{0.00}$
	LAC	Avila (12)	$0.93_{0.26}$	$0.90_{0.29}$	$1.26_{0.48}$	$0.93_{0.26}$
		CardDefault (2)	$1.25_{0.44}$	$1.25_{0.43}$	$1.87_{0.34}$	$1.25_{0.44}$
		Landsat (6)	$1.02_{0.25}$	$0.99_{0.22}$	$1.69_{0.93}$	$1.02_{0.25}$
		LetterRecognition (26)	$0.97_{0.32}$	$0.93_{0.29}$	$2.29_{1.65}$	$0.97_{0.32}$
		MagicGamma (2)	$1.07_{0.26}$	$1.06_{0.24}$	$1.63_{0.48}$	$1.07_{0.26}$
	APS	Avila (12)	$1.22_{0.69}$	$1.09_{0.61}$	$2.33_{1.37}$	$1.22_{0.69}$
		CardDefault (2)	$1.36_{0.50}$	$1.35_{0.50}$	$1.91_{0.29}$	$1.36_{0.50}$
		Landsat (6)	$1.32_{0.78}$	$1.27_{0.74}$	$2.27_{1.39}$	$1.31_{0.78}$
		LetterRecognition (26)	$2.49_{2.63}$	$2.20_{2.42}$	$6.80_{6.11}$	$2.49_{2.63}$
		MagicGamma (2)	$1.21_{0.49}$	$1.19_{0.49}$	$1.74_{0.44}$	$1.21_{0.49}$

C.4 Insights Ablation

We analyzed the dependence of the expected size of prediction sets on various user-specified parameters in Section 4.1. Here we empirically validate our analysis by providing experimental results on such parameter changes.

We use the same setup as the one highlighted in Section 6.1, and further add 3 settings, changing a user-specified parameter one at a time. These settings are: (i) an increase in the amount of training data from 25% to 50% of the dataset, hence learning a machine learning model that generalizes better (which is further used to implement the non-conformity function), (ii) a decrease in the significance level α from 0.1 to 0.01, allowing for fewer errors in the conformal system, and (iii) an increase in the amount of calibration data n from 25% to 50% of the dataset.

Table 8 illustrates the change in the marginal expected prediction set size (the Monte Carlo average) under varying user-specified parameters. We observe that: (i) an increase in the amount of training data decreases the expected prediction set size, (ii) a decrease in the significance level increases the expected prediction set size, and (iii) an increase in the calibration data does not affect the expected prediction set size by much; the only exception is when using the 0-1 loss non-conformity function on Landsat, but the standard deviations of the estimates are high under this setting. These experimental results empirically validate our analysis in Section 4.1.

C.5 Conditional Expected Prediction Set Size

Here we analyze the expected size of prediction sets conditioned on the test inputs. This is an extension of our experimental results in Section 6.3, where we considered the non-conformity functions CQR [Romano et al., 2019] and APS [Romano et al., 2020]; here we extend our analysis to other non-conformity functions as well.

Figure 2 depicts histograms of the Monte Carlo average and our point estimates for the expected set sizes conditioned on varying test inputs. Similar to our observations from Figure 1, the histograms in Figure 2 look identical for the two estimates, despite our point estimate not having seen the calibration data. The only exception is when using the 0-1 loss non-conformity function on Landsat. These results further corroborate the efficacy of our estimates in approximating the expected size of prediction sets conditioned on the test datum feature.

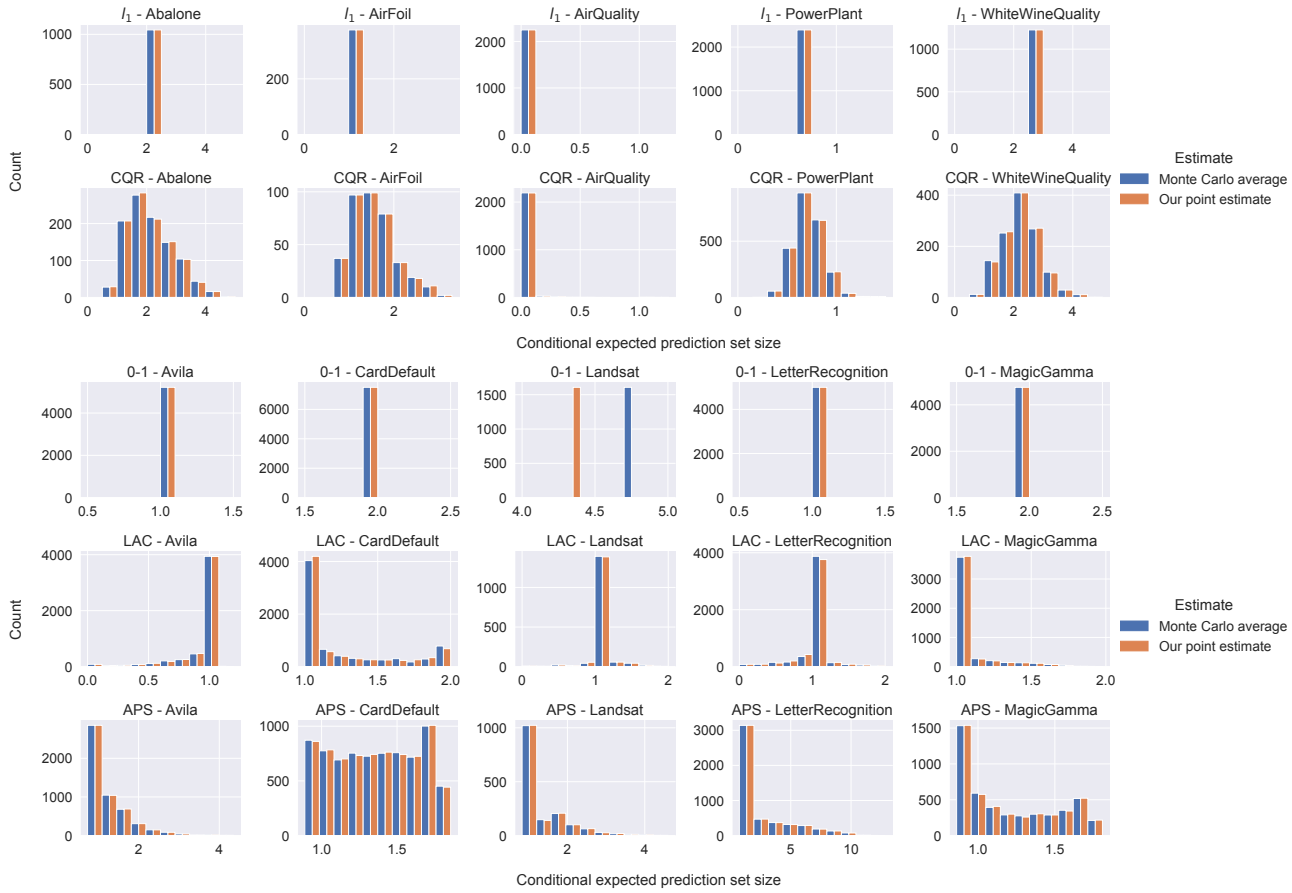

Figure 2: **Expected prediction set sizes conditioned on the test datum feature.** We illustrate the expected sizes of split conformal prediction sets conditioned on varying test datum features using different non-conformity scores (rows) and UCI datasets (columns). The estimates are obtained via Monte Carlo averaging and our point estimates (refer to the legend for the color scheme); they are depicted as a histogram with side-by-side bars.

C.6 Other Estimates

The Monte Carlo average is the commonly used empirical estimate for the expected prediction set size. In Section 6.2, we compared our estimates from Section 5 with the Monte Carlo average, where the former used $3\times$ fewer data points in its approximation. Here we compare the two when the same data points are used for both.

We first describe how the Monte Carlo average is obtained. This equates to sampling a (pseudo) calibration data and obtaining conformal prediction sets on multiple (pseudo) test data; the average size of the obtained prediction sets is the Monte Carlo average. As we did before, we assume access to k data points $Z'_1 = (X'_1, Y'_1), \ldots, Z'_k = (X'_k, Y'_k)$ that are available for the purpose of deriving estimates. When $k > n$, we can sample n data points to be the (pseudo) calibration data and the remaining $k - n$ data points to be the (pseudo) test data; this matches the number of calibration data used in the estimation and the one we want to estimate for. However, when $k \leq n$ (which is often the case), we cannot avoid the mismatch in the calibration data used in the estimation and the one we want to estimate for. Instead, we split the data into $k/2$ each for both calibration and test. As a result, the sizes of the obtained prediction sets are i.i.d. (which will be useful later) and we call their average the same-data Monte Carlo average (since we will use the same k accessible data points as the ones used for our estimates).

We follow the same setup as in Section 6.1 to compare our point estimate and the same-data Monte Carlo average with respect to the regular Monte Carlo average; Table 9 compares these estimates. We observe that the means of both our point estimates and the same-data Monte Carlo averages are close to that of the Monte Carlo average, but our point estimates have comparable or smaller standard deviations. When comparing the absolute errors between the individual estimates and the mean Monte Carlo average, the errors of our point estimates never exceed those of the same-data Monte Carlo averages. This corroborates the practical use of our point estimates.

Table 9: **Marginal expected prediction set sizes (point estimates).** We compare different point estimates for the marginal expected prediction set size with respect to regular Monte Carlo averaging. They are obtained via our point estimates and the same-data Monte Carlo average. We also compute the absolute errors between the individual estimates and the mean Monte Carlo average. We report the means and standard deviations.

			Marginal expected prediction set size			Absolute error	
		Dataset	Monte Carlo average	Our point estimate	Same-data Monte Carlo average	Our point estimate	Same-data Monte Carlo average
Regression ($\mathcal{Y} = \mathbb{R}$)	l_1	Abalone	$2.19_{0.09}$	$2.19_{0.09}$	$2.20_{0.13}$	$0.07_{0.05}$	$0.11_{0.08}$
		AirFoil	$1.39_{0.10}$	$1.39_{0.09}$	$1.41_{0.13}$	$0.08_{0.05}$	$0.10_{0.08}$
		AirQuality	$0.02_{0.00}$	$0.02_{0.00}$	$0.02_{0.00}$	$0.00_{0.00}$	$0.00_{0.00}$
		PowerPlant	$0.70_{0.02}$	$0.70_{0.02}$	$0.70_{0.02}$	$0.01_{0.01}$	$0.02_{0.01}$
		WhiteWineQuality	$2.58_{0.08}$	$2.58_{0.07}$	$2.58_{0.11}$	$0.06_{0.05}$	$0.09_{0.07}$
	CQR	Abalone	$2.17_{0.98}$	$2.16_{0.17}$	$2.18_{0.28}$	$0.15_{0.09}$	$0.27_{0.09}$
		AirFoil	$1.58_{0.64}$	$1.58_{0.07}$	$1.60_{0.11}$	$0.05_{0.04}$	$0.09_{0.07}$
		AirQuality	$0.02_{0.11}$	$0.02_{0.00}$	$0.02_{0.00}$	$0.00_{0.00}$	$0.00_{0.00}$
		PowerPlant	$0.73_{0.26}$	$0.73_{0.01}$	$0.74_{0.02}$	$0.01_{0.01}$	$0.01_{0.01}$
		WhiteWineQuality	$2.24_{0.89}$	$2.24_{0.10}$	$2.24_{0.10}$	$0.07_{0.06}$	$0.08_{0.06}$
Classification (discrete $\mathcal{Y}$)	0-1	Avila (12)	$1.00_{0.00}$	$1.00_{0.00}$	$1.00_{0.00}$	$0.00_{0.00}$	$0.00_{0.00}$
		CardDefault (2)	$2.00_{0.00}$	$2.00_{0.00}$	$2.00_{0.00}$	$0.00_{0.00}$	$0.00_{0.00}$
		Landsat (6)	$4.79_{2.14}$	$4.47_{1.31}$	$4.61_{2.24}$	$1.07_{0.82}$	$1.93_{1.16}$
		LetterRecognition (26)	$1.00_{0.00}$	$1.00_{0.01}$	$1.00_{0.00}$	$0.00_{0.01}$	$0.00_{0.00}$
		MagicGamma (2)	$2.00_{0.00}$	$2.00_{0.00}$	$2.00_{0.00}$	$0.00_{0.00}$	$0.00_{0.00}$
	LAC	Avila (12)	$0.93_{0.26}$	$0.93_{0.01}$	$0.93_{0.01}$	$0.00_{0.00}$	$0.01_{0.01}$
		CardDefault (2)	$1.25_{0.44}$	$1.25_{0.01}$	$1.25_{0.02}$	$0.01_{0.01}$	$0.02_{0.01}$
		Landsat (6)	$1.02_{0.25}$	$1.02_{0.02}$	$1.02_{0.03}$	$0.01_{0.01}$	$0.02_{0.02}$
		LetterRecognition (26)	$0.97_{0.32}$	$0.97_{0.01}$	$0.97_{0.02}$	$0.01_{0.00}$	$0.01_{0.01}$
		MagicGamma (2)	$1.07_{0.26}$	$1.07_{0.01}$	$1.07_{0.02}$	$0.01_{0.01}$	$0.01_{0.01}$
	APS	Avila (12)	$1.22_{0.69}$	$1.22_{0.02}$	$1.22_{0.03}$	$0.02_{0.01}$	$0.03_{0.02}$
		CardDefault (2)	$1.36_{0.50}$	$1.36_{0.01}$	$1.36_{0.02}$	$0.01_{0.01}$	$0.02_{0.01}$
		Landsat (6)	$1.32_{0.78}$	$1.32_{0.03}$	$1.31_{0.05}$	$0.03_{0.02}$	$0.04_{0.03}$
		LetterRecognition (26)	$2.49_{2.63}$	$2.49_{0.07}$	$2.50_{0.10}$	$0.05_{0.04}$	$0.08_{0.06}$
		MagicGamma (2)	$1.21_{0.49}$	$1.21_{0.01}$	$1.21_{0.02}$	$0.01_{0.01}$	$0.02_{0.01}$

Table 10: **Marginal expected prediction set sizes (interval estimates).** We compare interval estimates for the marginal expected prediction set size (with $\gamma = 0.1$). They are obtained via our interval estimates, the central limit theorem (CLT), Hoeffding's inequality (HI), and Bernstein's inequality (BI). We compute their sizes and error frequencies in bounding the mean Monte Carlo average. We report the means and standard deviations.

			Interval size				Interval error frequency			
		Dataset	Ours	CLT	HI	BI	Ours	CLT	HI	BI
Regression ($\mathcal{Y} = \mathbb{R}$)	l_1	Abalone	$0.84_{0.10}$	$0.00_{0.00}$			0.00	1.00		
		AirFoil	$0.91_{0.11}$	$0.00_{0.00}$			0.00	1.00		
		AirQuality	$0.01_{0.00}$	$0.00_{0.00}$			0.00	1.00		
		PowerPlant	$0.13_{0.01}$	$0.00_{0.00}$			0.00	1.00		
		WhiteWineQuality	$0.70_{0.07}$	$0.00_{0.00}$			0.00	1.00		
	CQR	Abalone	$0.64_{0.03}$	$0.13_{0.01}$			0.00	1.00		
		AirFoil	$0.73_{0.12}$	$0.15_{0.01}$			0.00	0.48		
		AirQuality	$0.00_{0.00}$	$0.01_{0.00}$			1.00	0.12		
		PowerPlant	$0.10_{0.01}$	$0.02_{0.00}$			0.00	0.52		
		WhiteWineQuality	$0.03_{0.08}$	$0.12_{0.01}$			0.89	0.52		
Classification (discrete $\mathcal{Y}$)	0-1	Avila (12)	$0.00_{0.00}$	$0.00_{0.00}$	$0.58_{0.00}$	$0.02_{0.00}$	0.00	0.00	0.00	0.00
		CardDefault (2)	$0.00_{0.00}$	$0.00_{0.00}$	$0.08_{0.00}$	$0.00_{0.00}$	0.00	0.00	0.00	0.00
		Landsat (6)	$4.95_{0.21}$	$0.00_{0.00}$	$0.52_{0.00}$	$0.03_{0.00}$	0.00	1.00	1.00	1.00
		LetterRecognition (26)	$5.50_{5.85}$	$0.00_{0.00}$	$1.27_{0.00}$	$0.04_{0.00}$	0.00	0.00	0.00	0.00
		MagicGamma (2)	$0.03_{0.06}$	$0.00_{0.00}$	$0.10_{0.00}$	$0.00_{0.00}$	0.00	0.00	0.00	0.00
	LAC	Avila (12)	$0.05_{0.00}$	$0.02_{0.00}$	$0.58_{0.00}$	$0.04_{0.00}$	0.00	0.44	0.00	0.10
		CardDefault (2)	$0.12_{0.01}$	$0.02_{0.00}$	$0.08_{0.00}$	$0.04_{0.00}$	0.00	0.62	0.07	0.42
		Landsat (6)	$0.15_{0.01}$	$0.03_{0.00}$	$0.52_{0.00}$	$0.06_{0.00}$	0.00	0.58	0.00	0.26
		LetterRecognition (26)	$0.08_{0.01}$	$0.02_{0.00}$	$1.27_{0.00}$	$0.06_{0.00}$	0.00	0.50	0.00	0.07
		MagicGamma (2)	$0.09_{0.01}$	$0.02_{0.00}$	$0.10_{0.00}$	$0.03_{0.00}$	0.00	0.56	0.00	0.39
	APS	Avila (12)	$0.14_{0.01}$	$0.05_{0.00}$	$0.58_{0.00}$	$0.08_{0.00}$	0.01	0.48	0.00	0.22
		CardDefault (2)	$0.12_{0.01}$	$0.03_{0.00}$	$0.08_{0.00}$	$0.04_{0.00}$	0.00	0.54	0.08	0.36
		Landsat (6)	$0.24_{0.03}$	$0.09_{0.00}$	$0.52_{0.00}$	$0.15_{0.01}$	0.00	0.36	0.00	0.11
		LetterRecognition (26)	$0.51_{0.04}$	$0.17_{0.01}$	$1.27_{0.00}$	$0.28_{0.01}$	0.00	0.40	0.00	0.18
		MagicGamma (2)	$0.11_{0.01}$	$0.03_{0.00}$	$0.10_{0.00}$	$0.05_{0.00}$	0.00	0.41	0.01	0.20

Furthermore, since the prediction set sizes are i.i.d., we can obtain confidence intervals for the expected prediction set size using concentration inequalities; we make use the following ones: (i) the central limit theorem (CLT),

(ii) Hoeffding's inequality (HI), and (iii) Bernstein's inequality (BI). CLT is valid only asymptotically. HI is finite-sample valid for random variables with known bounds, so is useful only for classification problems (where the set size is bound by $[0, |\mathcal{Y}|]$). BI is finite-sample valid for random variables with known bounds and variances (useful only for classification problems); however, since the variance is unknown and needs to be approximated, the intervals may not be valid confidence intervals. Alternatively, our interval estimates combine the expected set size in Theorem 1 and the Dvoretzky–Kiefer–Wolfowitz inequality [Dvoretzky et al., 1956; Massart, 1990].

We again follow the same setup as in Section 6.1 to compare our interval estimates and the ones obtained using CLT, HI, and BI; Table 10 compares these estimates (with γ set to 0.1). We compute the interval sizes and the frequencies of error in bounding the mean Monte Carlo average. We observe that the CLT and the BI intervals are small, but their error frequencies can be high. On the other hand, our intervals and the HI intervals consistently achieve errors below the requirement of $\gamma = 0.1$. Further, the HI intervals cannot be applied to regression problems, and they do not change with the non-conformity function as the intervals are determined by the classification problem (the number of labels/classes), the number of data points used in the approximation, and γ. On the other hand, our intervals can be applied to regression problems, and they adapt with the non-conformity function being used, often resulting in smaller interval sizes. Hence our estimated intervals provide practical use.

D EXPERIMENTS ON SYNTHETIC EXAMPLE

We experimentally validate our theoretical results for the marginal expected prediction set size in Theorem 1 and Corollary 3 through a synthetic example. We design a setup where the distribution of the calibration non-conformity scores is known. We set the space of non-conformity scores to be a discrete set of m values $\mathcal{R} = \{r_1, \ldots, r_m\}$, where $r_1 < \ldots < r_m$. We set the distribution of the calibration non-conformity scores to be a beta-binomial distribution $\text{BetaBin}(m-1, a, b)$ over the indices, with parameters $a, b > 0$. Then, for a non-conformity score $r_i \in \mathcal{R}$, $\tilde{P}_R(r_i) = \mathbb{P}\{\text{BetaBin}(m-1, a, b) \leq i-2\}$. Additionally, we set $\#_R(r) = 2$.

For a fixed set of values of the parameters a, b, m, n, and $\alpha = 0.1$, we theoretically compute the expected prediction set size using Equation (5). We compare this against: (i) the average size of prediction sets constructed by running the conformal prediction algorithm (the Monte Carlo average), and (ii) our point estimates and confidence intervals (for a given γ) of the expected prediction set size computed using Section 5.1. We randomly sample n i.i.d. non-conformity scores; we use these as the calibration non-conformity scores for the Monte Carlo average, and use the same as accessible non-conformity scores for our estimates. We repeat the process 10 times using different random seeds. Additionally, we vary the parameter values in the following way: $a, b \in \{.0625, .25, 1, 4, 16\}$, $m, n \in \{10, 100, 1000, 10000\}$, and $\gamma \in \{0.1, 0.01\}$. This results in a total of 800 settings, repeated 10 times each.

Figure 3 plots the theoretically expected prediction set sizes and their empirical estimates across these different settings, which we average over different a and b to obtain line plots. The identity line (dashed black line) is the theoretically expected prediction set size from Equation (5). We make the following observations. (i) The Monte Carlo average (solid black line) collapses to the identity line as n increases from left to right; this validates our quantification of the expected set size in Theorem 1. (ii) The average of our point estimates (green line) also collapses to the identity line as n increases. (iii) Our confidence intervals contain the identity line with high probability; for $\gamma = 0.1$ (orange lines) and $\gamma = 0.01$ (blue lines), the confidence intervals contain the theoretically expected size values 99.9% and 100.0% of the time respectively. Additionally, as n increases, these confidence intervals collapse to the identity line. These validate our estimates in Section 5.1 and our result in Corollary 3.

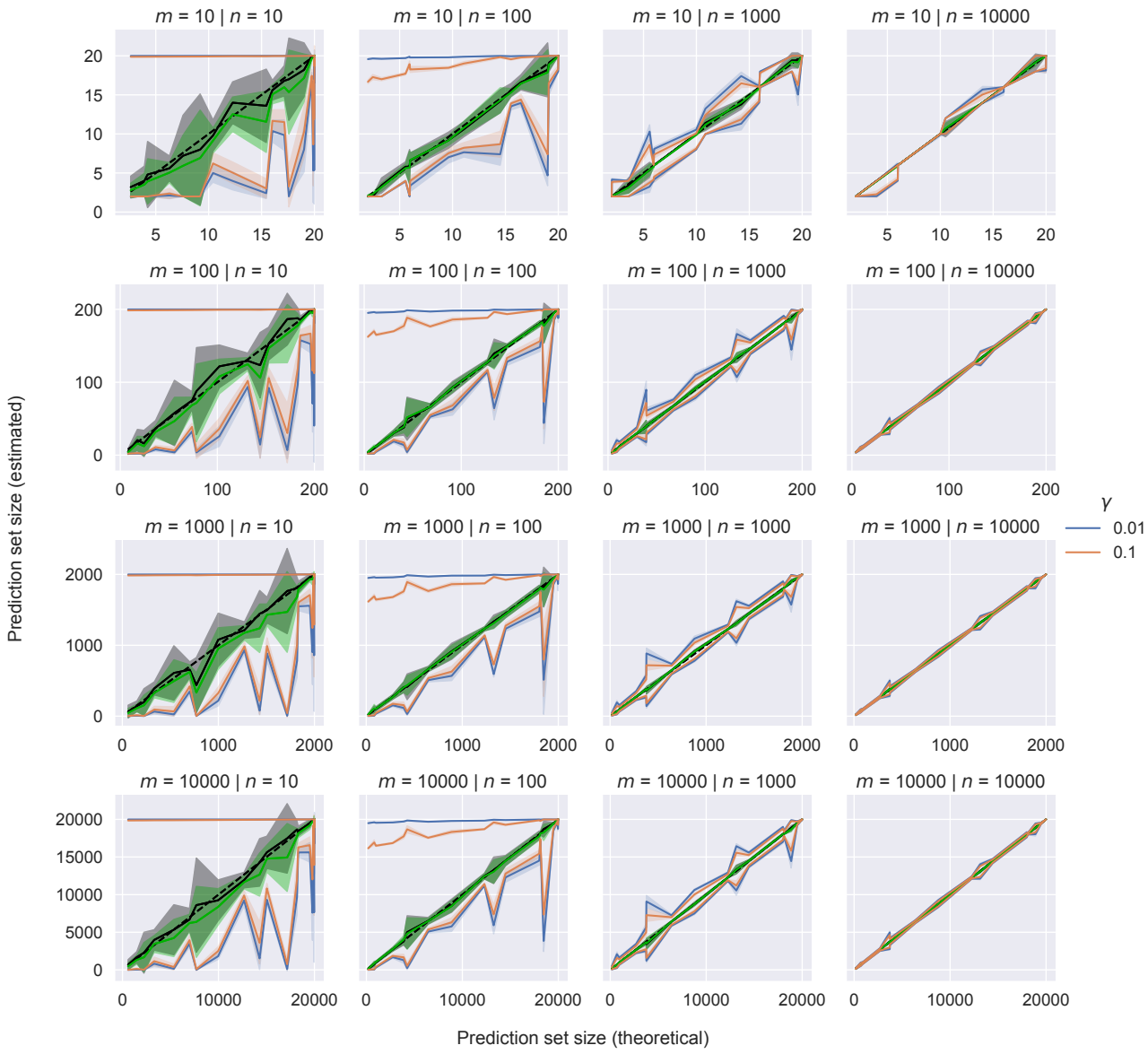

Figure 3: **Marginal expected prediction set sizes (synthetic example).** We plot the theoretically expected prediction set sizes (cf. Equation (5)) on the x-axis vs. its empirical estimates on the y-axis. These estimates include: (i) the Monte Carlo average (solid black line), (ii) our point estimate from Section 5.1 (green line), and (iii) our upper-lower confidence bounds from Corollary 3 (orange/blue lines, changing with γ as per the legend). α is set to 0.1 and the results are averaged to a line plot over different a and b values (bands denote the standard deviations). Additionally, the dashed black line is the identity line. The number of calibration data points n increases from left to right. The size of the space of non-conformity scores m increases from top to bottom.